Лабораторная работа 02

ПСКП

**Задание 01**

1. Разработайте серверное приложение **02-01**, которое на запрос [**http://localhost:5000**](http://localhost:5000)**/html** возвращает ответ, в теле которого содержится файл **index.html** (расположен в той же папке, что и скрипт **02-01.js**).
2. Файл **index.html** содержит статическую HTML-разметку с данными студента (факультет, курс, группа, фамилия, имя и отчество).
3. С помощью браузера проверьте работоспособность приложения.

**Задание 02**

1. Разработайте серверное приложение **02-02**, которое на GET-запрос [**http://localhost:5000**](http://localhost:5000)**/png** возвращает ответ, в теле которого содержится файл **pic.png** (расположен в той же папке, что и скрипт **02-02.js**).
2. Файл **pic.png** содержит любое изображение в формате png.
3. С помощью браузера проверьте работоспособность приложения.

**Задание 03**

1. Разработайте серверное приложение **02-03**, которое на GET-запрос [**http://localhost:5000/api/name**](http://localhost:5000/api/name)возвращает ответ, в теле которого содержится ваша фамилия имя и отчество в текстовом виде (text/plain).
2. C помощью браузера проверьте работоспособность приложения.
3. Проверьте работоспособность приложения с помощью **POSTMAN**.

**Задание 04**

1. Разработайте серверное приложение **02-04**, которое на запрос к [**http://localhost:5000/xmlhttprequest**](http://localhost:5000/xmlhttprequest)возвращает ответ, в теле которого содержится файл **xmlhttprequest.html** (расположен в той же папке, что и скрипт **02-04.js**).
2. Файл **xmlhttprequest.html** содержит HTML-разметку и JS.
3. С помощью объекта **XMLHTTPRequest** выполняется асинхронный GET-запрос к [**http://localhost:5000/api/name**](http://localhost:5000/api/name) иполученный ответ (фамилия, имя и отчество) динамически встраивается в HTML-разметку страницы.

**Задание 05**

1. Разработайте серверное приложение **02-05**, которое на запрос к [**http://localhost:5000/fetch**](http://localhost:5000/fetch)возвращает ответ, в теле которого содержится файл **fetch.html** (расположен в той же папке, что и скрипт **02-05.js**).
2. Файл **fetch.html** содержит HTML-разметку и JS.
3. С помощью объекта **fetch** выполняется асинхронный GET-запрос к [**http://localhost:5000/api/name**](http://localhost:5000/api/name)иполученный ответ (фамилия, имя и отчество) динамически встраивается в HTML-разметку страницы.

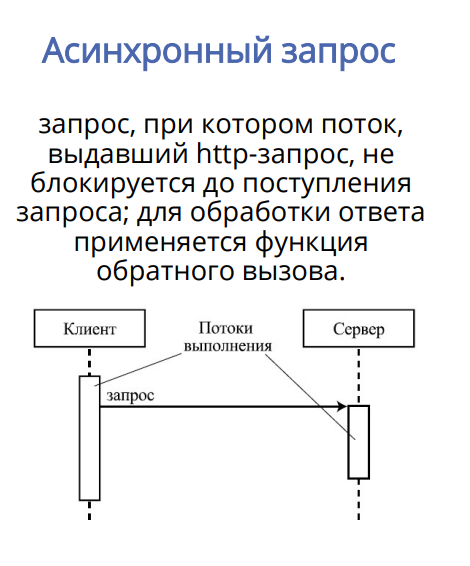
**Задание 06**

1. Разработайте серверное приложение **02-06**, которое на запрос к [**http://localhost:5000/jquery**](http://localhost:5000/jquery)возвращает ответ, в теле которого содержится файл **jquery.html** (расположен в той же папке, что и скрипт **02-06.js**).
2. Файл **jquery.html** содержит HTML-разметку и JS.
3. С помощью **jQuery.ajax** (функция библиотеки jQuery, обеспечьте в **jquery.html** соответствующий <script> c jQuery CDN: [https://code.jquery.com](https://code.jquery.com/)) выполняется асинхронный GET-запрос к [**http://localhost:5000/api/name**](http://localhost:5000/api/name)иполученный ответ (фамилия, имя и отчество) динамически встраивается в HTML-разметку страницы.

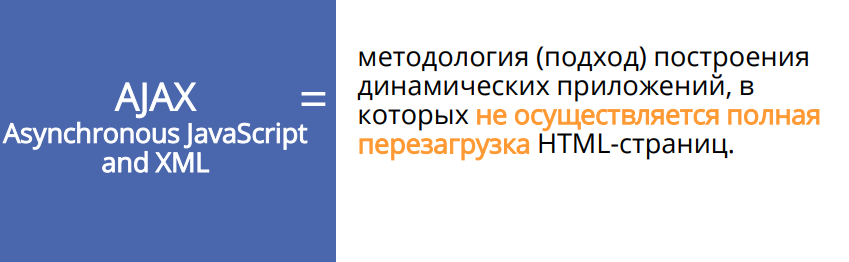
**Задание 07.** Ответьте на следующие вопросы.

1. Что такое **асинхронный запрос**?
2. Что такое **AJAX**?
3. Поясните назначение HTTP-заголовка **Content-Type**.
4. Поясните назначение функции **require**.
5. Поясните понятие «**Модуль Node.js**».
6. Поясните понятие «**Node.js built-in modules**» («**Node.js Core modules**»).
7. Какой модуль **NODE.JS** обеспечивают работу с протоколом HTTP, с файловой системой?
8. Что такое **npm**?
9. Что такое **асинхронный запрос**?

Асинхронный запрос (или асинхронный HTTP-запрос) - это запрос, который выполняется без блокирования основного потока выполнения программы. Вместо того, чтобы ждать завершения запроса перед продолжением работы, программа может выполнять другие задачи или события, пока запрос обрабатывается.

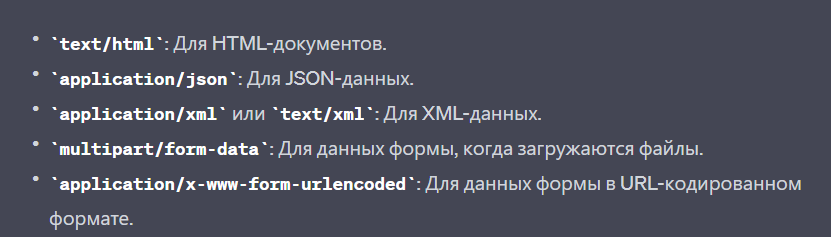


1. Что такое **AJAX**?



1. Поясните назначение HTTP-заголовка **Content-Type**.

HTTP-заголовок **Content-Type** используется для определения типа данных, который содержится в теле HTTP-сообщения. Этот заголовок сообщает получателю о том, как интерпретировать или обрабатывать данные, которые были переданы в запросе или ответе.



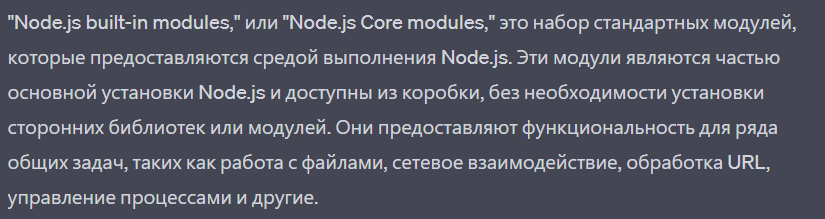
1. Поясните назначение функции **require**.

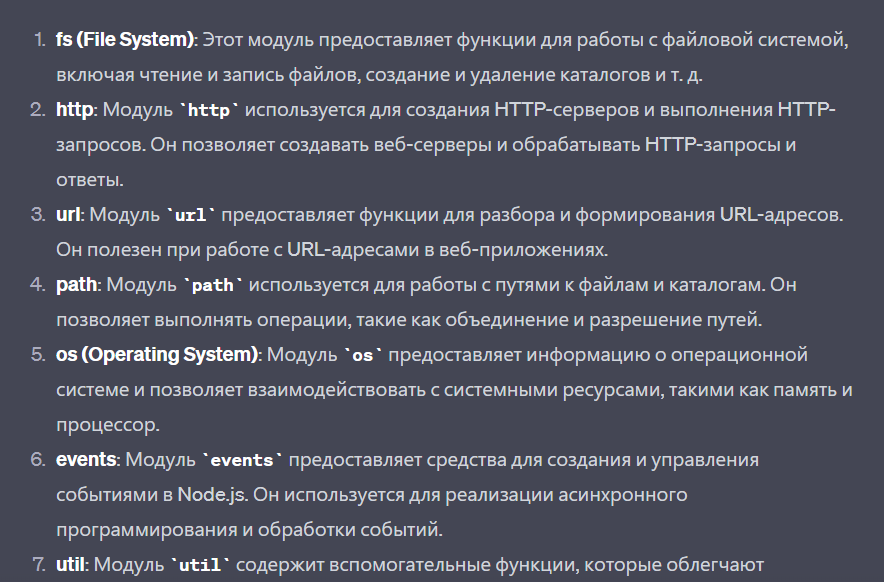
Назначение функции **require** состоит в том, чтобы обеспечить модульность и организацию кода в Node.js, позволяя разбивать код на небольшие, самодостаточные модули, которые можно затем использовать в других частях вашего приложения.

1. Поясните понятие «**Модуль Node.js**».

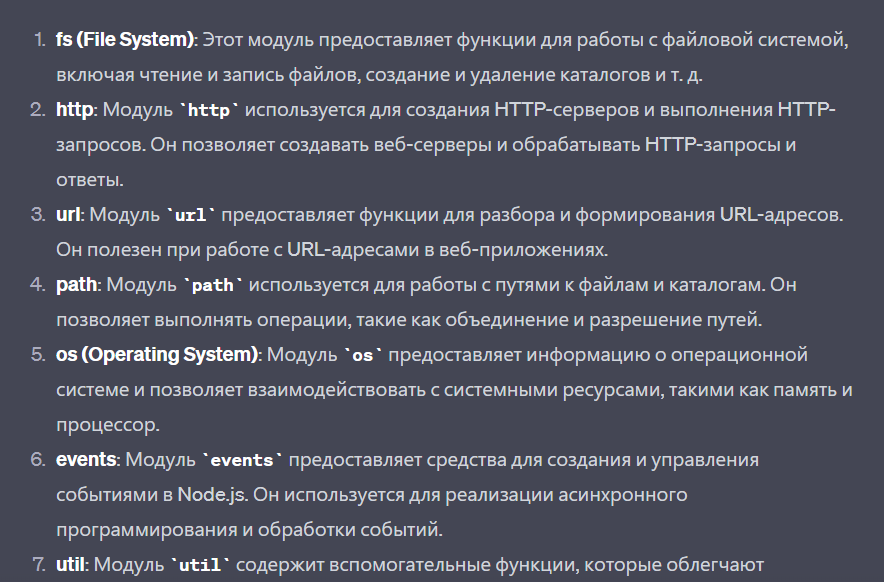
В Node.js модуль - это самодостаточная и независимая единица кода, которая может содержать переменные, функции и даже объекты или классы. Модули в Node.js созданы для организации кода и обеспечения его модульности, что позволяет легко разбивать большие приложения на более мелкие и управляемые компоненты.

1. Поясните понятие «**Node.js built-in modules**» («**Node.js Core modules**»).





1. Какой модуль **NODE.JS** обеспечивают работу с протоколом HTTP, с файловой системой?



1. Что такое **npm**?

npm (Node Package Manager) - это стандартный менеджер пакетов (библиотек и модулей) для среды выполнения Node.js. Он предоставляет средства для установки, управления и обновления пакетов JavaScript, которые могут быть использованы в Node.js-приложениях.