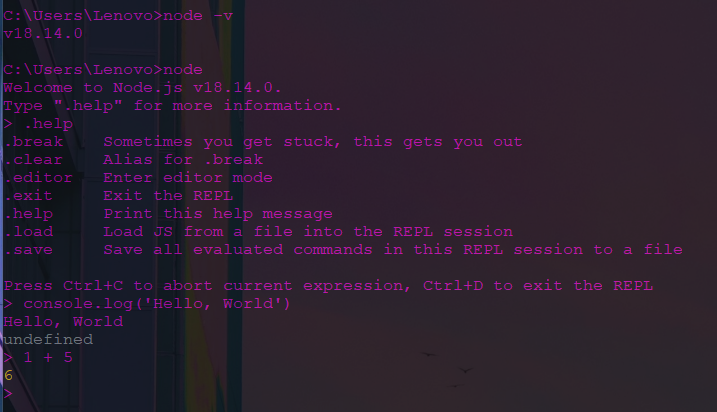
Лабораторная работа 01

ПСКП

ПОИТ-3

**Задание 01**

1. Установите Node.js.
2. Проверьте работоспособность Node.js.



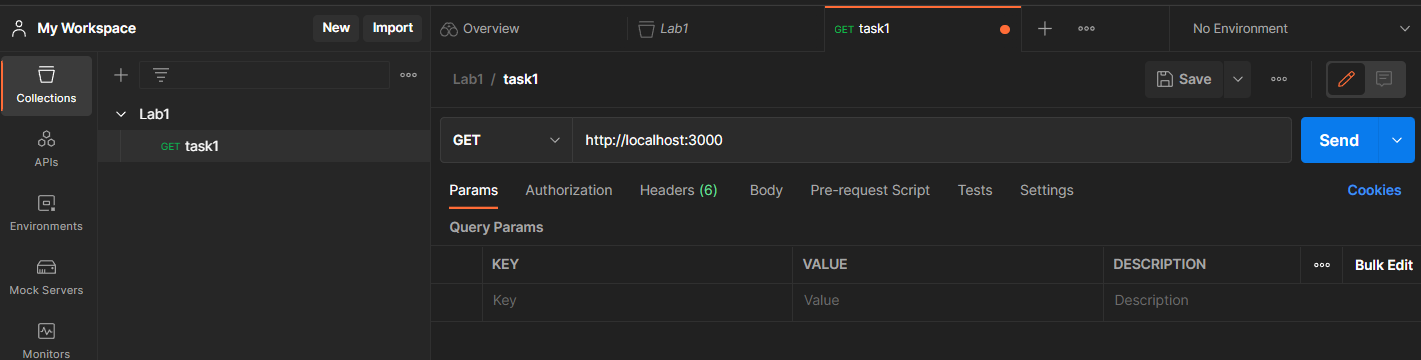
**Задание 02**

1. Разработайте серверное приложение **01-01**, использующее модуль http и возвращающее ответ с разметкой ***<h1>Hello World</h1>*.**
2. Проверьте работоспособность приложения с помощью браузера.

<http://localhost:3000>



1. Проверьте работоспособность приложения с помощью приложения ***POSTMAN***.



**Задание 03**

1. Разработайте серверное приложение **01-02**, на основе разработанного в задании 2, которое в ответе сервера пересылает html-страницу с содержимым запроса (метод, uri, версия протокола, заголовки, тело).
2. Выполните GET-запрос к серверу с помощью браузера и убедитесь в работоспособности сервера.
3. Выполните POST-запрос (содержащий данные в теле сообщения) к серверу с помощью POSTMAN и убедитесь в работоспособности сервера.

**Задание 04.** Ответьте на следующие вопросы.

В Node.js **модуль** — это набор функций и объектов JavaScript, который могут использовать внешние приложения.

1. **Дайте определение понятию «Интернет».**

**Интернет (или Internet**) - это глобальная система взаимосвязанных компьютерных сетей, которая использует набор интернет-протоколов (TCP/IP) для связи между сетями и устройствами.

* сеть на основе TCP/IP;
* стандарты Internet (RFC, STD);
* службы Интернет (DNS, SMTP/POP3/IMAP, WWW, FTP, Telnet, SSH,…);
* организации, управляющие сетью Internet (ISOC, IETF, W3C, ICANN, IANA, …).

1. **Дайте определение понятию «Служба Интернет».**

Это системы, предоставляющие услуги пользователям Интернета. К ним относятся: электронная почта, WWW, телеконференции, списки рассылки, FTP, IRC, а также другие продукты, использующие Интернет как среду передачи информации.

1. **Дайте определение понятию «Узел сети Интернет».**

Устройство, имеющее IP-адрес и подключенное сети Интернет (обычно к сети Интернет провайдера). Каждый узел характеризуется своей программно аппаратной платформой – аппаратурой и операционной системой.

1. **Дайте определение понятию «клиент-серверное приложение».**

Приложение (программа) с клиент-серверной архитектурой:

• состоит из двух компонент – клиента и сервера;

• клиент и сервер взаимодействуют между собой в соответствии с заданными правилами (протоколами);

• для взаимодействия между клиентом и сервером в соответствии с правилами (протоколом) должно быть установлено соединение;

• инициатором соединения – клиент.

1. **Дайте определение понятию «сетевой протокол».**

Набор правил и соглашений, описывающие взаимодействие компьютеров в сети.

1. **Перечислите основные свойства протокола HTTP.**

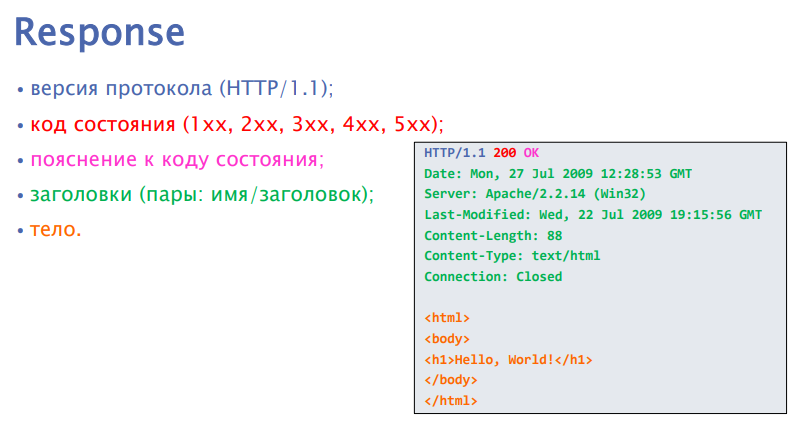
**HTTP**(HyperText Transfer Protocol - протокол передачи гипертек­ста) - протокол прикладного уровня стека протоколов TCP/IP. предна­значенный для передачи данных по сети с использованием транспорт­ного протокола TCP.

* версии HTTP/1.1 – действующий (текстовый), HTTP/2 – действующий (бинарный), HTTP/3 (раньше HTTP-over-QUIC, стадия черновика);
* два типа абонентов: клиент и сервер;
* два типа сообщений: request и response;
* от клиента к серверу – request;
* от сервера к клиенту – response;
* на один request всегда один response, иначе ошибка;
* одному response всегда один request, иначе ошибка;
* stateless;
* TCP-порты: 80, 443;
* для адресации используется URI или URL;
* поддерживается W3C, описан в нескольких RFC

1. **Перечислите состав информации, пересылаемой в HTTP-запросе.**



1. **Перечислите состав информации, пересылаемой в HTTP-ответе.**



1. **Перечислите группы заголовков HTTP.**

**General**: общие заголовки, используются в запросах и ответах.

**Request**: используются только в запросах.

**Response**: используются только в ответах

**Entity**: для сущности в ответах и запросах

1. **Дайте определение понятию «web-приложение».**

Клиент-серверное приложение, у которого клиент и сервер взаимодействуют по протоколу HTTP. Когда говорят о разработке web-приложения, говорят о разработке frontend (клиента) и backend (сервера)

1. **Дайте определение понятиям «frontend» и «backend».**

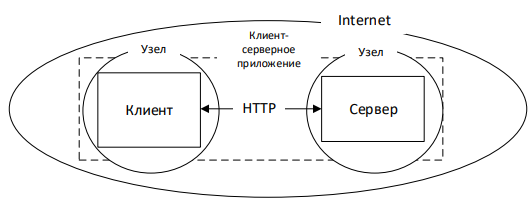
Фронтенд — видимая часть цифрового продукта, бэкенд — его программно-аппаратная составляющая, скрытая от глаз юзера.

Во фронтенде пишется код, который работает в браузере и определяет то, что в итоге увидит юзер. Бэкенд — это тоже код, но он нужен для сервера, где обрабатываются запросы пользователей.

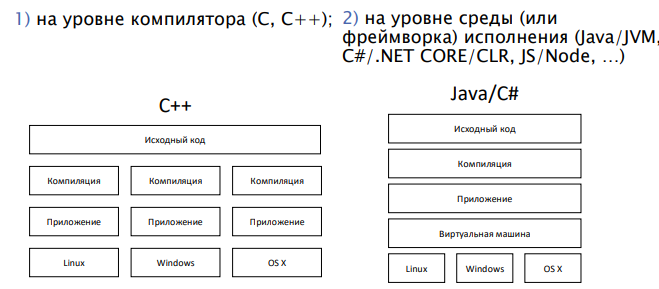
1. **Дайте определение понятию «кроссплатформенное приложение».**

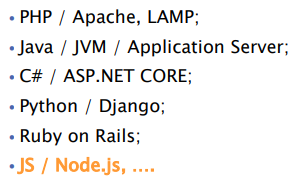
приложение, способное работать на более чем одной программно-аппаратной (аппаратура + операционная система) платформе

1. **Изобразите и поясните общую схему web-приложения.**



1. **Назовите основные технологии разработки серверных кроссплатформенных приложений.**





1. **Поясните понятие «асинхронная операция».**

запрос, при котором поток, выдавший http-запрос, не блокируется до поступления запроса; для обработки ответа применяется функция обратного вызова.

1. **Поясните основное назначение NODE.JS.**

Программная платформа для разработки серверных web-приложений на языке JS/V8.

1. **Перечислите основные свойства NODE.JS.**

* основан на **Chrome V8**;
* **среда (контейнер) исполнения** приложений на JavaScript;
* поддерживает механизм **асинхронности**;
* ориентирован на **события**;
* **однопоточный** (код приложения исполняется только в одном потоке, один стек вызовов); обычно в серверах для каждого соединения создается свой поток, в Node.js все соединения обрабатываются в одном JS-потоке;
* **не блокирует** выполнение кода при вводе/выводе (в файловой системе до 4х одновременно);
* в состав Node.js входят инструменты: **npm** – пакетный менеджер; **gyp** - Python-генератор проектов; **gtest** – Google фреймворк для тестирования С++ приложений;

использует библиотеки: **V8** – библиотека V8 Engine, **libuv** – библиотека для абстрагирования неблокирующих операций ввода/вывода (представляет собой обертку над epoll, kqueue, IOCP); **llhttp** – легковесный парсер http-сообщений (написан на C и не выполняет никаких системных вызовов); **c-ares** -библиотека для работы с DNS; **OpenSSL**