



# Основы Программирования

Занятие №11  
Сеть Интернет.  
Сервер.



## IT Education Academy

[WWW.ITEA.UA](http://WWW.ITEA.UA)

# План занятия

- Интернет, сеть
- IP адрес, доменное имя
- Сервер, хостинг
- Передача данных
- Сервер (программное обеспечение)

# Компьютерная сеть

---

Это система, которая обеспечивает передачу данными между вычислительными устройствами (компьютерами, смартфонами и тд.)

Для передачи данных используются разные среды - проводные и беспроводные.



# Интернет

---

Это всемирная система объединенных компьютерных сетей, предназначенная для хранения и передачи информации.

# IP адрес

---

Уникальный адрес устройства в рамках сети.

Локальный IP адрес - уникальный адрес в рамках локальной сети.

Глобальный IP адрес - уникальный в рамках глобальной сети.

Состоит из 4х чисел от 0 до 255: 0.0.0.0 - 255.255.255.255.

Необходим для поиска конкретного компьютера в сети.

Т.е. зная IP адрес компьютера, можно передать ему данные.

# Взаимосвязь сетей

---

Подключаясь к Интернету вы пользуетесь услугами интернет провайдера.

Интернет провайдер (ИП) - это организация имеющая непосредственный доступ к Интернету (подключена к другим компьютерам в рамках глобальной сети).

Когда вы хотите передать данные какому-то компьютеру в сети (указывая его IP адрес), на самом деле ваши данные передаются ИП, а он, зная кому их передать, отправляет получателю.

Часто, ИП не связан напрямую с вашим получателем, между ними может быть несколько устройств. Тогда ИП передает ваши данные другому устройству в сети и так далее, пока ваши данные не дойдут до получателя.

# Взаимосвязь сетей

---

Узнать сколько различных устройств проходят ваши данные на пути к серверу Гугл.

Для этого открыть консольное окно (cmd).

И написать команду:

**tracert google.com**

# Доменное имя

---

Пользоваться IP адресами для взаимодействия с другими компьютерами не очень удобно - их сложно запоминать.

Никто бы не стал пользоваться Гуглом, если бы его нужно было открывать как 172.217.20.174.

Вместо этого используются доменные имена - символьный адрес устройства.

Существуют специальные службы DNS - сервера, которые хранят в себе табличку соответствий

google.com - 172.217.20.174

facebook.com - 185.60.216.35

и тд.

Когда вы делаете передачу данных, указывая доменное имя - вы просите вашего ИП обратиться к этой службе и узнать IP адрес сервера (найти соответствие).



# Сервер, хостинг

---

Сервер - это обычный компьютер, который имеет доступ к Интернету.

Как и все другие компьютеры он может принимать данные и передавать их.

Однако, сервер намного более мощный - часто сотни ядер процессоров и сотни гигабайт оперативной памяти.

Хостинг - это услуга по предоставлению части вычислительных ресурсов сервера в пользование клиенту.

Т.е. купив хостинг, вы получаете возможность запустить на сервере свою программу.

# Свой сайт

---

Для того, чтобы сделать доступным свой сайт другим пользователям нужно\*:

1. Купить хостинг.
2. Расположить на хостинге свой сайт (т.е. загрузить его на сервер).
3. Купить доменное имя.
4. Связать доменное имя с адресом хостинга.

\* - имеется ввиду что вы имеете рабочую версию сайта (написанный html, css, js, и другие языки).

# Передача данных

---

Как уже говорилось, компьютер работает только с 0 и 1. Соответственно при передачи данных используются различные способы кодировки 0 и 1.

Передавая данные от одного устройства к другому по кабелю:

1. Ваш компьютер, с помощью сетевой карты, кодирует 0 и 1 в электрический ток с соответствующими характеристиками.
2. Этот ток течет по кабелю и доходит до другого устройства.
3. В другом устройстве его сетевая карта считывает ток и расшифровывает его в 0 и 1, запуская программу которая обработает эти данные.

# Передача данных

---

Ваши данные передаются не как постоянный и непрерывный набор 0 и 1.

Любая передача данных разбивается на пакеты определенного размера.

Например, передавая данные размером 1MB ваш компьютер разобьет этот файл на пакеты(например) по 64 байта (или больше) и сделает вместо 1й передачи 16тыс передач.

# Пакет

---

Пакет данных помимо самих данных содержит другую информацию, например адрес получателя, размер пакета и тд. Эта информация используется устройствами для того, чтобы переправлять пакеты дальше по сети.

# Порт

---

Для того, чтобы сетевая карта знала какую именно программу ей запустить, существует такое понятие как порт.

Порт это число от 1 до 65535.

Каждый пакет содержит в себе порт на который он был отправлен.

Когда вы запускаете программу и хотите чтобы она получала данные с сетевой карты вы «подписываетесь» на событие сетевой карты, указывая какой порт вас интересует.

Когда на сетевую карту придут данные, она «оповестит» вашу программу.

В браузере, по умолчанию, указывается порт 80.

Т.е. если вы хотите, чтобы ваши клиенты не указывали порт, ваш сайт должен подписаться на порт 80.



# Порт

---

Удостовериться, что по умолчанию порт - 80.

Открыть в браузере `google.com:80`

# Сервер (ПО)

---

В области программного обеспечения также существует понятие сервера. Это программа, которая подписана на порт, т.е. ожидает данные и реагирует на них.

Входные данные - называются запросами клиента. Во входных данных содержится информация о том, что интересует клиента, например html страничка, или запись в базе данных.

Реагируя на запрос, сервер также посылает данные клиенту - ответ на запрос, например html страничку или запись в базе данных.



# Сервер в Node.js

---

```
var http = require('http');

function processRequest(request, response) {
  response.writeHead(200, {'Access-Control-Allow-Origin': '*'});
  response.write('hello world');
  response.end();
}

var server = http.createServer(processRequest);

server.listen(8080);
console.log("Server started at 8080 port");
```

# Сервер в Node.js

---

Чтобы сделать запрос на наш сервер (ПО) с помощью браузера нужно указать адрес компьютера на котором он запущен и порт на котором он запущен.

Чтобы сказать браузеру сделать запрос на ваш же компьютер используйте адрес 127.0.0.1.

Порт на котором запущен наш сервер - 8080.

Сделайте запрос в браузере на наш сервер.

# Сервер в Node.js

---

Сделайте запрос на сервер вашего соседа.

Для этого нужно узнать его адрес:

попросите соседа открыть командную строку (cmd) и выполнить команду  
**ipconfig**

И сказать его IPv4 адресс.

Сделайте запрос на его сервер.

# Сервер в Node.js

```
C:\Users\Vlad>ipconfig
```

```
Настройка протокола IP для Windows
```

```
Адаптер беспроводной локальной сети Подключение по локальной сети* 1:
```

```
Состояние среды. . . . . : Среда передачи недоступна.
```

```
DNS-суффикс подключения . . . . . :
```

```
Адаптер беспроводной локальной сети Подключение по локальной сети* 2:
```

```
Состояние среды. . . . . : Среда передачи недоступна.
```

```
DNS-суффикс подключения . . . . . :
```

```
Адаптер беспроводной локальной сети Беспроводная сеть:
```

```
DNS-суффикс подключения . . . . . :
```

```
Локальный IPv6-адрес канала . . . : fe80::98aa:b643:d255:5ba0%4
```

```
IPv4-адрес. . . . . : 192.168.0.103
```

```
Маска подсети . . . . . : 255.255.255.0
```

```
Основной шлюз. . . . . : 192.168.0.1
```

# Запрос на сервер в JS

---

```
<button onclick="makeRequest()">Make request</button>
<script>
  function reactToResponse(response) {
    if (response.readyState === 4 && response.status === 200) {
      alert(response.responseText);
    }
  }
  function makeRequest() {
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function() {
      reactToResponse(this);
    };
    xhttp.open("GET", "http://127.0.0.1:8080", true);
    xhttp.send();
  }
</script>
```

# Запрос на сервер в JS

---

Добавить input для ввода адреса сервера.

По нажатию на кнопку делать запрос на указанный адрес.

# Summary

---

1. Сеть - это система для передачи данных между компьютерами.
2. Интернет - это объединение сетей.
3. IP адрес и доменное имя связаны между собой с помощью DNS сервера.
4. Сервер (с т.з. аппаратного обеспечения) - это компьютер, сервер (с т.з. программного обеспечения) - это программа, реагирующая на запросы и отправляющая ответы.
5. Порт - нужен сетевой карте для оповещения сервера (ПО).
6. Данные передаются не целиком, а разбиваются на пакеты.
7. В пакетах содержится дополнительная информация.

# Домашнее задание

---

1. Создать сервер на Node.js (порт 8080).

Создать массив имен (до 5 штук). На каждый запрос возвращать случайное имя (найти код из предыдущих занятий getRandom(min, max)).

2. Создать страничку на html с кнопкой “Make request”. По нажатию на которую делать запрос на сервер (http://127.0.0.1:8080). На ответ сервера добавлять новый параграф (элемент p) на страничку с содержимым ответа.

Т.е. ваш код должен получать случайное имя с сервера и добавлять его на страничку.



**Спасибо за внимание!**