

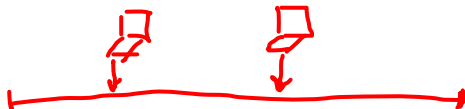
31.01.2015

## Сети

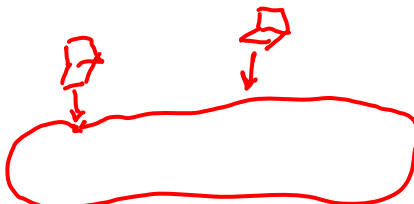
LAN - Local Area Network

Wireless - Wi-Fi, Mobile Networks (EDGE, HSPA, 3G, 4G), Laser optics

Топология  
Шина (Bus)



Кольцо (Ring)



Звезда (Star)



Коммутатор, бывают его виды: маршрутизатор, хаб (hub), свитч (switch)

-----  
Адресация в сети

IP-адрес (Internet Protocol) - уникальный адрес компьютера в сети.

IP (IPv4) - на номер компьютера отводится 4 байта ( $2^{32}$  адресов)

IPv6 - на номер отводится 128 бит

Адрес состоит из 4-х байтов, включающих номер компьютера в сети, а также номер сети.

Запись представляется в виде 4 чисел (0..255) в десятичном виде через .

192.168.30.83

Маска позволяет отделить номер компьютера от номера сети.

255.255.255.0

В маске содержится 1 и 0. 1 - показывают какие биты в адресе отвечают за номер сети, а 0 - за номер компьютера

11111111.11111111.11111111.00000000

номер сети

номер компьютера

Выделяют 3 уровня сетей:

- Сеть категории А

по битам 0<7 бит на номер сети 0..127><24 бита на номер компьютера 1..16777214>

8.8.8.8 - адрес DNS-сервера Google

- Сеть категории В

по битам 10<14 бит номер сети от 128.0 до 191.255><16 бит на номер компьютера 1..65534>

- Сеть категории С

по битам 110<21 бит номер сети от 192.0.0 до 223.255.255><8 бит номер компа 1..254>

номера компьютеров 0 и последний в своей сети зарезервированы. Например, адрес с последним номером является широковещательным.



192.168.30.255

По типу отношений компьютеров в сети

- одноранговые сети (peer-to-peer), сети без выделенного сервера
- server-based - с выделенным сервером

DHCP - dynamic host configuration protocol

При автоматической выдаче адресов ОС автоматически получает ip 169.254.0.0 с маской 255.255.0.0

-----

	Web browser	The Internet	195.70.197.2	ФС сервера
1	http://school30.spb.ru			
2		Кто отвечает за school30.spb.ru?		
3	Запрос на 195.70.197.2			
4			Запрос на файл index.html	
5				Забрать содержимое index.html
6			Отправить содержимое ответа	
7	Показать полученные данные			

Ответ о том, кто отвечает на запросы по доменному имени school30.spb.ru формируется системой доменных имен (DNS).

## HTML

1986-1991

HTML - hyper text markup language - гипертекстовый язык разметки.

Идея - документ "размечается" специальными пометами, тэги. Заголовки, параграфы, мета-данные (данные о данных - автор, название и т.д.)

<ИМЯ\_ТЭГА>...</ИМЯ\_ТЭГА>

Тэги структуры документа

```
html
  head
    title
  body
    h1,...,h6
    p
    b (bold), strong, i (italic), em, u, sup, sub
    ul - нумерованный, ol - нумерованный списки
    li - элемент списка
    table
```



thead, tbody, tfooter  
tr - строка  
td - ячейка  
img - вставка изображения  
a - гиперссылка (anchor)

DOM - Document Object Model

```
<ИМЯ_ТЭГА  
ИМЯ_АТТРИБУТА1=ЧИСЛОВОЕ_ЗНАЧЕНИЕ_АТТРИБУТА  
ИМЯ_АТТРИБУТА2="ЗНАЧЕНИЕ_АТТРИБУТА"  
ИМЯ_АТТРИБУТА3='ЗНАЧЕНИЕ_АТТРИБУТА'>  
...  
</ИМЯ_ТЭГА>
```

Атрибуты для таблиц:

table -  
cellpadding - внутренний отступ от границы ячейки (ед.измерение)  
cellspacing - отступ между ячейками (ед.измерение)  
width - ширина (ед.измерение)  
border - ширина рамки (ед.измерение)

(ед.измерение - это 5px 2cm 50% 14pt 0.5em)

td  
rowspan - склейка строк  
colspan - склейка столбцов

ul  
type - определяет формат вывода нумерации disk, circle, square  
ol  
type - 1, i, I, a, A

img  
src - источник изображения (относительный путь к изображению либо URI - Uniform Resource Identifier)  
width - ширина (ед.измерения)  
height - высота (ед.измерения)

a  
href - цель ссылки (URI)  
target - место назначения ("\_blank")

-----  
Задание на дом

1. как работает электронная почта? Каким образом определяется на какой почтовый сервер отправлять почту по адресу info@school130.spb.ru?

DNS файл зоны MX

2. Сделать страничку о себе. ФИО, класс, фото, адрес эл.почты в виде корректной ссылки

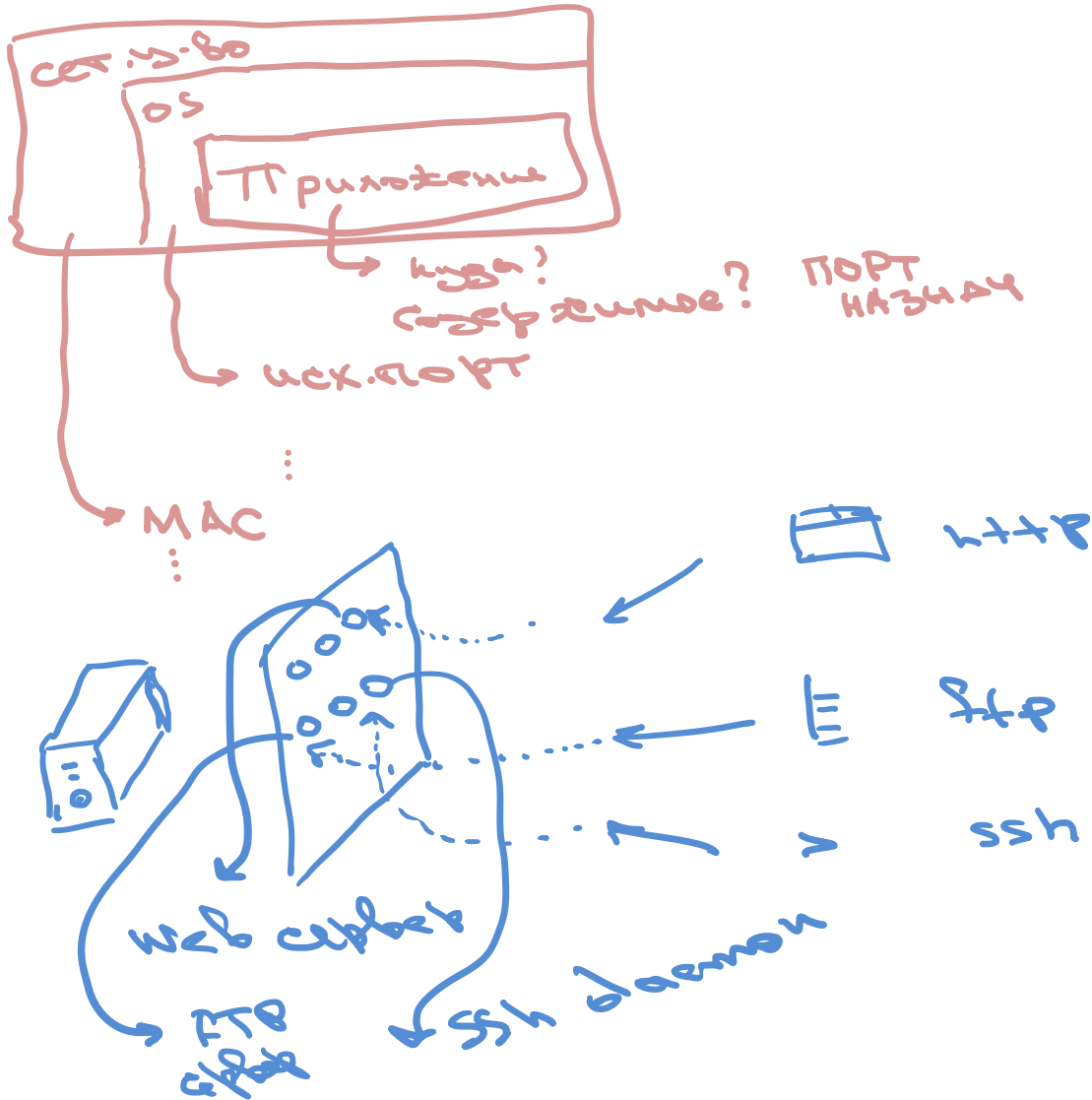
W3C - World Wide Web Consortium

w3schools.com

07.02.2015

http запрос содержит в себе доменное к которому обращается и является частью TCP/IP

TCP/IP - Transfer Control Protocol / Internet Protocol



Порт - это нумерованное место для приложения, обслуживающего запросы.

Порт	Протокол	Приложение
№ 80	- http-запросы	- веб-сервер
22	- ssh	- демон удаленного управления ssh



## Веб-сервера

apache

WAMP - Windows + Apache + MySQL + PHP

LAMP - Linux + Apache + MySQL + PHP

EasyPHP (easyphp.org)

c:\easyphp

<file:///z:/www/1.html>

<http://127.0.0.1/1.html>

	Web browser	The Internet	195.70.197.2	ФС сервера	Server-side script
1	<a href="http://school30.spb.ru">http://school30.spb.ru</a>				
2		Кто отвечает за school30.spb.ru?			
3	Запрос на 195.70.197.2				
4			Запрос на файл index.php		
				Вызывает интерпретатор php	
					php index.php
6			Результат выполнения как текст отправляется в браузер		
7	Показать полученные данные				

## PHP

<http://php.net/docs.php>

Скриптовый язык, схожий с языком Си

Переменные

```
$x = 3;
```

```
$foo = "bar"; // в переменную с именем foo присвоить строку
```

```
$bar = 3;
```

```
echo $bar; // вывести на экран
```

```
$$foo = 7;
```



```
echo $bar;
```

Арифметические операции

```
+ - / *
```

Строковые операции

```
. - операция конкатенации
```

Циклы

```
while( условие )  
    команда
```

Составная команда

```
{  
}
```

```
for( ин-р счетчика; условие; изменение счетчика )  
    команда
```

do

```
{  
    команда  
} while( условие );
```

```
foreach( $array as [ $key => ] $value )  
    команда
```

Типы данных

строки

```
-- литералы "abc" 'def'
```

числовые типы данных

```
-- int, float, double
```

массивы

```
$a = array(1, 2, '4', "мама!");
```

вывести массив

```
print_r($a);
```

вывод переменной

```
var_dump($a);
```

Ассоциированный массив - массив пар "ключ"- "значение"

Ключ - это любая величина (не объект) - число, строка

Значение - это любой объект

```
$a = array(1, 2, array(1, 2, '4', "мама!"), "мама!");
```

Шаблон проектирования "модель" - "представление"

```
-- обработка данных - в одном месте
```

```
-- отображение - в другом
```

Функции

```
function ИМЯ(параметр, параметр)
```

```
{
```

```
    ...
```



```
    return ...;
}
```

### Задание

Телефонная книжка

Состоит из записей, каждая из которых

- Ф
- И
- Телефон
- имя файла с фотографией

1. EasyPHP
2. header.php, footer.php
3. data.php - файл с данными
4. functions.php - файл с функциями
  - print\_human\_row - для списка человек
  - print\_human\_table - для страницы одного человека
5. index.php - страница, содержащая таблицу списка человек, каждая из которых является ссылкой <a href=...> на страницу человека
6. person.php - страница с информацией о человеке

```
<table>
  <tr>
    <td><a href="person.php      ">Иван Мария</a></td>
  </tr>
</table>
```

---

Передача параметров в страницу методом GET - через параметры URL

index.php?a=b&c=4&r=&q=5

? - разделяет имя скрипта от набора параметров

& - разделяет параметры

параметры - это пары ключ-значение через =

а значение "b"

с            4

r

q            5

Суперглобальный массив \$\_GET - содержит параметры запроса в виде ассоциированного массива

В функциях все переменные - локальные. Чтобы обратиться к глобальным переменным, необходимо их "объявить" с помощью global

```
$str = "ABC";
```

```
function print_abc()
{
    global $str; // если этого не сделать, то будет ошибка времени исполнения

    echo $str;
}
```



14/02/2015

## СУБД MySQL

СУБД - Система Управления Базами Данных

База Данных - это способ хранения данных, обеспечивающий быстрый доступ, поиск и представление данных.

СУБД - программа.

Виды БД

- объектные БД (MongoDB)
- реляционные БД (relation - отношение) (OracleSQL, MS SQL, PostgreSQL, Oracle MySQL, MS Office Access(?))
- встраиваемые БД (пары ключ-значение) (BerkleyDB - библиотека на C)

### MySQL

Реляционная БД - это такой способ организации структур данных, в котором выделяются сущности, характеризующиеся атрибутами (полями) и отношениями с другими сущностями.

Каждая сущность может быть представлена в виде таблицы, где колонки - это атрибуты, а строки - это различные значения сущности.

Например, таблица оценок ученика. - атрибуты - ученик + день

- значения - ФИО ученика + его оценки

### Нормальная форма БД

1. 1 НФ - Не должно быть дублирования данных по столбцам.

- Автор
- Автор2
- Название книги

Иванов И.		Петров П.		Избранное из сумасшедшего дома
Иванов И.				Приколы сумасшедшего дома

-> перевод в 1 НФ

Создать сущность Автор, а для каждой книги указать ее авторов

Авторы:

Id - идентификатор (целое число), уникальное значение

1		Иванов И
2		Петров П

### Книжки

1		2		Избранное из сумасшедшего дома
1				Приколы сумасшедшего дома

2 НФ. Не должно быть повторяющихся данных по строкам

Избыточность данных

### Виды отношений

1 - 1 - один к одному - это отношение оформляется в виде атрибута, т.е. колонки в таблице сущности

1 - М - один к много - такой вид отношений оформляется в виде указания ссылки на одинарную сущность

Папа - Сын (таблица пап, таблица детей, которые ссылаются на своего папу)





М - М - много ко много - такой вид отношений оформляется через третью таблицу  
Ученики - Кружки (таблица учеников, таблица кружков и таблица ученики-кружки).

Общая схема работы с СУБД

1. Подключиться к СУБД (логин-пароль)
2. Выбрать активную БД
3. Запросы
4. Отключиться от СУБД

#### 1. Подключиться к СУБД

В консоли

1. {EasyPHP}\mysql\bin\mysql.exe -u root -p  
клиентское приложение
2. Ввести пароль (пароль по умолчанию пустой)

#### 2. Выбрать активную БД

mysql> USE DATABASE ИМЯ\_БД; - выбирает активную БД  
SHOW DATABASES - показывает все БД в СУБД  
SHOW TABLES - показывает таблицы активной БД

#### 3. Запросы

SQL - Structured Query Language - Структурированный Язык Запросов.

Идея - осуществлять запросы, обладающие определенной структурой и результаты запросов - это таблицы.

#### **Создание**

-- БД

```
CREATE DATABASE ИМЯ_БД;
```

```
mysql> CREATE DATABASE school;
```

-- таблицы

```
CREATE TABLE ИМЯ_ТАБЛИЦЫ (  
    ОПРЕДЕЛЕНИЕ КОЛОНОК,  
    ...  
);
```

ОПРЕДЕЛЕНИЕ КОЛОНКИ ::=

```
ИМЯ_КОЛОНКИ ТИП_КОЛОНКИ МОДИФИКАТОРЫ |  
PRIMARY KEY(ИМЯ КОЛОНКИ) |
```

#### **ТИПЫ:**

INT, DOUBLE, TEXT, CHAR, CHAR[M] - строка длиной M символов, DATE, DATETIME

#### **МОДИФИКАТОРЫ:**

NOT NULL - значение в колонке не может быть пустым. Если при вставке в таблицу значение пустое, то вставка не произойдет - будет вызвана ошибка.

AUTO\_INCREMENT - автоматически увеличивающееся значение

### Пример

```
CREATE TABLE human (  
  id      INT NOT NULL AUTO_INCREMENT,  
  surname TEXT NOT NULL,  
  name    TEXT NOT NULL,  
  PRIMARY KEY(id)  
);  
  
CREATE TABLE phones (  
  id      INT NOT NULL AUTO_INCREMENT,  
  phone   TEXT NOT NULL,  
  human_id INT NOT NULL,  
  PRIMARY KEY(id)  
);
```

### Изменение

Alternation - изменение

ALTER ЧТО КАК

```
ALTER TABLE human ADD COLUMN age INT DEFAULT 0;
```

DEFAULT - значение по-умолчанию. Если при вставке значение не указано, будет внесено значение по-умолчанию

### Выборки

**SELECT ЧТО FROM ОТКУДА УСЛОВИЯ ГРУППИРОВКИ ОГРАНИЧЕНИЯ**

SELECT \* FROM humans; - выбирает все колонки для всех записей из таблицы humans  
SELECT name FROM humans;

ЧТО - колонка или функция

### УСЛОВИЯ

WHERE УСЛОВИЕ1 [AND | OR] УСЛОВИЕ2

УСЛОВИЕ - сравнение (числовое: > < >= <= = - сравнение на равенство <> - сравнение на неравенство, на значение NULL - IS NULL / IS NOT NULL)

```
SELECT * FROM humans WHERE age IS NOT NULL AND id > 15;
```

- строковое сравнение

= <>

LIKE - сравнение по шаблону % - любое количество любых символов, ? -

один любой символ

```
SELECT surname FROM humans WHERE name LIKE '%an%';
```

ILIKE - сравнение по шаблону без учета регистра

- логическое

```
SELECT * FROM humans WHERE (SELECT COUNT(id) > 0 FROM phones WHERE  
human_id=humans.id); - выбрать всех людей, у которых количество телефонов>0
```



## ОГРАНИЧЕНИЯ

LIMIT число1 OFFSET число2  
ограничивает выдачу не более число1 строк со смещением от первой выдачи на число2 строк

```
SELECT * FROM humans WHERE age > 12 LIMIT 2 OFFSET 3; - 4,5 записи
```

## ОТКУДА

```
WHERE ИМЯ_T1 [, ИМЯ_T2]
```

```
SELECT name, phone FROM humans, phones;
```

```
1 Иван И  
2 Петр П
```

```
1 21 1  
2 22 1  
3 11 2
```

```
Иван 21  
Иван 22  
Иван 11  
Петр 21  
Петр 22  
Петр 11
```

Сочетание всех строк одной таблицы со всеми строками другой таблицы - декартово произведение.

```
SELECT name, phone FROM humans, phones WHERE human_id = humans.id;  
1 2  
3
```

```
FROM ИМЯ_T1 JOIN ИМЯ_T2 ON УСЛОВИЕ  
[JOIN ИМЯ_T3 ON УСЛОВИЕ]
```

```
SELECT name, phone FROM humans, phones WHERE human_id = humans.id;  
<=>  
SELECT name, phone FROM humans  
JOIN phones ON human_id = humans.id;
```

## ГРУППИРОВКИ

Все колонки, участвующие в условиях группировки должны быть в списке выводимых колонок.

```
GROUP BY КОЛОНКА[, КОЛОНКА]
```

```
SELECT name, COUNT(phones.id)  
FROM humans  
JOIN phones ON humans.id = phones.human_id  
GROUP BY human_id;
```

Выводит имя персоны и количество телефонов для данной персоны



## ВСТАВКА

```
INSERT INTO ИМЯ_ТАБЛИЦЫ [ (КОЛОНКА1, КОЛОНКА2, ...) ]  
VALUES (ЗНАЧЕНИЕ_ДЛЯ_КОЛОНКИ1, ...)[, (ЗНАЧЕНИЕ_ДЛЯ_КОЛОНКИ1, ...)];
```

```
INSERT INTO phones VALUES (1, '123'), (2, '234'), (1, '234');
```

```
INSERT INTO ИМЯ_ТАБЛИЦЫ [ (КОЛОНКА1, КОЛОНКА2, ...) ]  
SELECT .... FROM ...;
```

## УДАЛЕНИЕ

```
-- строк
```

```
DELETE FROM ИМЯ_ТАБЛИЦЫ [WHERE УСЛОВИЕ];
```

```
DELETE FROM pupils WHERE age < 10;
```

```
-- таблиц
```

```
DROP TABLE [IF EXISTS] ИМЯ_ТАБЛИЦЫ;
```

```
-- базы данных
```

```
DROP DATABASE [IF EXISTS] ИМЯ_БД;
```

## ОБНОВЛЕНИЕ ДАННЫХ

```
UPDATE ИМЯ_ТАБЛИЦЫ SET  
ИМЯ_КОЛОНКИ = ЗНАЧЕНИЕ [,ИМЯ_КОЛОНКИ = ЗНАЧЕНИЕ]  
WHERE УСЛОВИЕ;
```

```
UPDATE pupils SET surname = 'Lutkin' WHERE id=1;
```

```
UPDATE ИМЯ_ТАБЛИЦЫ SET  
(ИМЯ_КОЛОНКИ1, ИМЯ_КОЛОНКИ2...) = (ЗНАЧЕНИЕ1, ЗНАЧЕНИЕ2 ...)  
WHERE УСЛОВИЕ;
```

---

21/02/2015

```
$x = 1;
```

```
$y = "1";
```

```
if ($x == $y)  
echo "TRUE";
```

```
else  
echo "FALSE";
```

```
if ($x === $y) -- осуществляется проверка на совпадение типов  
echo "TRUE";
```

```
else  
echo "FALSE";
```