
Web-технологии

2-й год обучения

Преподаватель

Капитанов Андрей Иванович

ско@kardx.ru

2015 г.

Расписание

- Воскресенье
- **12:00** — 13:00
- 13:10 — **14:30**
- 814 офис

WT-2

1 семестр

Коллоквиум (~ 5-е занятие)

Контрольные + тестирования

Подготовка к конференции

Экзамен

2 семестр

Контрольные + тестирования

Олимпиада

Подготовка к курсовой работе

Контрольные + тестирования

Конференция

Коллоквиум

Сдача курсовой работы

Экзамен

Занятие #1

- **Выберите верный php тэг**

1) `<%php... %>`

2) `<#php... #>`

3) `?... ?`

4) `<?php... ?>`

2 / 12

- Какой результат выполнения следующего кода:

```
<?php
    $a = 500;
    $b = 200;
    echo $a % 2 * $b;
?>
```

- Какие из приведённых в коде условий отобразят значения?

```
<?php
$a = true; $b = false;

if ($a xor $b) { print "A"; }
if ($a or $b) { print "B"; }
if ($a and $b) { print "C"; }
?>
```

4 / 12

- **Как определить константу?**

- `constant('FOO', 'BAR');`
- `defineconstant('FOO', 'BAR');`
- `define('FOO', BAR);`
- `variable('FOO', 'BAR');`

5 / 12

- **Какая разница между равенством и эквивалентностью?**

- 1) Знак эквивалентности работает лишь для строк, а знак равенства применим к любым типам.
- 2) Знак равенства проверяет лишь значения операндов, а знак эквивалентности значения и их типы.
- 3) Никакой разницы нет.
- 4) Знак эквивалентности проверяет лишь значения операндов, а знак равенства значения и их типы.

6 / 12

- **Что будет выведено на экран в результате выполнения этой программы**

```
<?php
    $a = 15.75;
    echo (int) $a;
?>
```

- **Какая функция в РНР позволяет найти длину строки?**
 - strtok()
 - strpos()
 - str_repeat()
 - strlen()

8 / 12

- Какой результат выполнения следующего кода:

```
<?php  
    echo 30 * 5.7;  
?>
```


9 / 12

- **Какой тип данных не допустим в PHP?**
 - 1) string
 - 2) integer
 - 3) float
 - 4) char
 - 5) array
 - 6) object

10 / 12

- Какой из следующих операторов используется для конкатенации строк?

1) &

2) .

3) |

4) ->

5) =

6) +

11 / 12

- **Что будет выведено на экран в результате выполнения этой программы**

```
<?php  
    $a = -1;  
    echo ++$a;  
?>
```

12 / 12

- Является ли переменная **\$221BakerStreet** валидной?

Принципы работы Интернета



Статистика

- **2000 г.** – 361 млн. пользователей
 - **2014 г.** – 2,8 млрд. пользователей
 - **2015 г.** – 3,2 млрд. пользователей
-
- В среднем по миру **43 человек из 100** являются интернет пользователями.

Терминология

Сервер — любой отдельно взятый компьютер в Интернете, который позволяет другим машинам использовать себя в качестве "посредника" при передаче данных





Терминология

Порт — это некоторое число (от 0 до 65535) которое идентифицирует программу, желающую принимать данные из Интернета.

Терминология

- **Узел (хост)** — любая сущность, *имеющая уникальный IP-адрес* в Интернете.
- **Виртуальный хост** — это *хост, не имеющий уникального IP-адреса* в Сети, но, тем не менее, доступный указанием какого-нибудь дополнительного адреса (*например, его DNS-имени*).

Терминология

- **Провайдер** — организация, имеющая несколько модемных входов, к которым могут подключаться пользователи для доступа в Интернет.
- **Хостинг-провайдер** — это организация, которая может создавать хосты (виртуальные или обычные) в Интернете и продавать их различным клиентам, обычно за определенную плату.

Терминология

- **Хостинг** — это услуги, которые предоставляют клиентам хостинг-провайдеры.

Терминология

- **Протокол** — это набор правил, который позволяет системам, взаимодействующим в рамках сети, обмениваться данными в наиболее удобной для них форме.

Терминология

- **TCP** (Transmission Control Protocol)
Протокол управления передачей данных
- **HTTP** (Hypertext Transfer Protocol)
Протокол передачи гипертекста
базирующийся на **TCP**.

Терминология

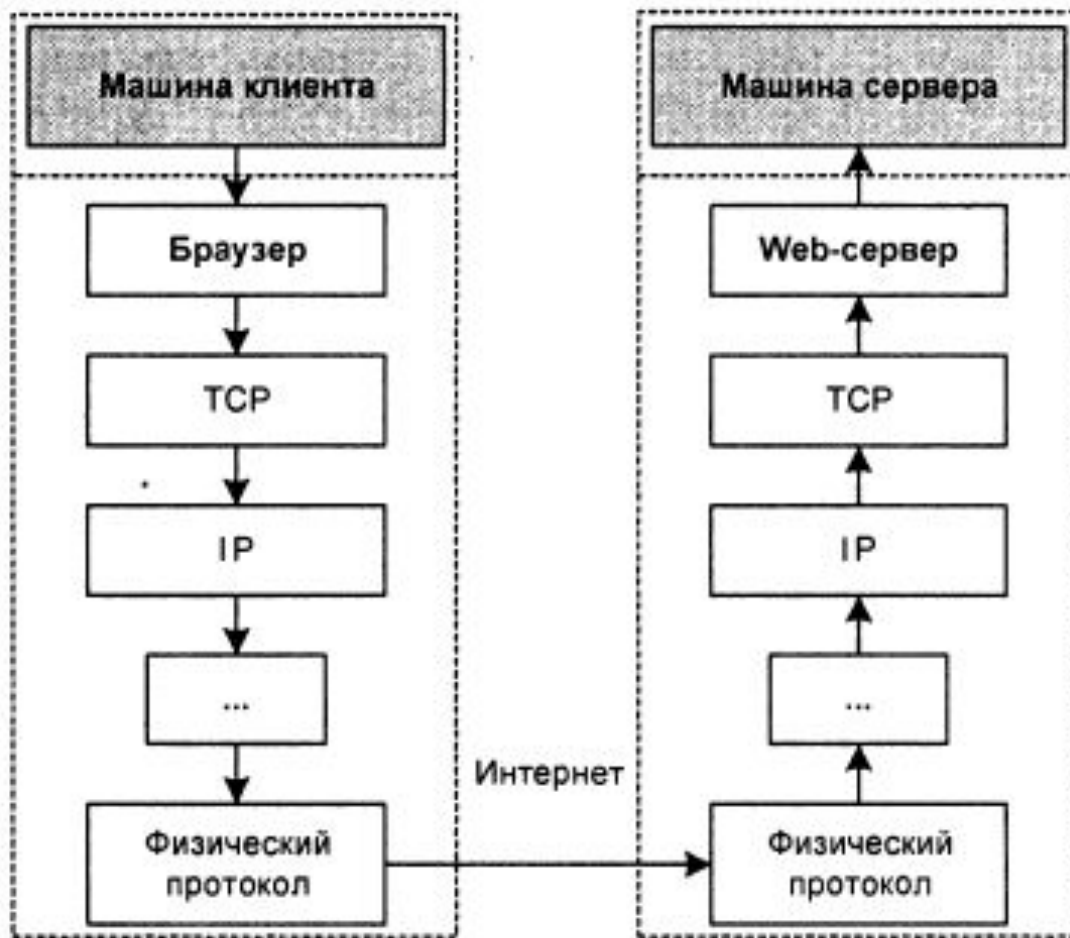
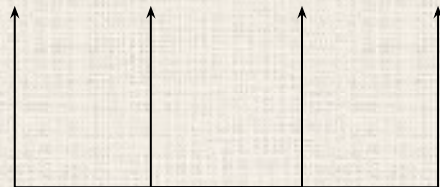


Рис. 1.1. Организация обмена данными в Интернете

IP-адрес

- Пример IP – адреса:
– **83.220.237.14**



(числа от 0 до 255)

Доменное имя

- Чтобы облегчить простым пользователям работу с Интернетом, придумали систему DNS
- **DNS** (Domain Name System) служба имен доменов.

Структура DNS

← DNS записи
← Доменные сервера

mail.yahoo.com

www.yahoo.com

login.yahoo.com

yahoo.com

Записи, типа
www.yandex.ru, также
сервера зон третьего
уровня

Сервера отвечающие за
домены второго уровня
типа mail.ru, yandex.ru ...

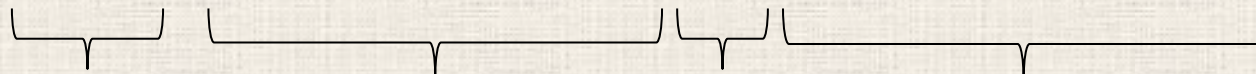
TLD DNS сервера
отвечают за зоны
первого уровня
типа ru, org, com, net...

Корневые (root) DNS
сервера

URL

- URL (Universal Resource Locator)
универсальный локатор ресурса
- **Пример:**

`http://example.com:80/path/doc.html`



Протокол

Имя хоста

Порт

Путь к странице

Протокол HTTP

Стартовая строка

В запросе:

МЕТОД URI HTTP/**Версия**

Пример:

GET /wiki/ HTTP/1.0

Протокол HTTP

Стартовая строка

В ответе:

HTTP/**Версия** **Код_состояния** **Пояснение**

Пример:

HTTP/**1.0** **200** **OK**

404

**NOT
FOUND**

404

NOT
FOUND

NOT
FOUND

Классы кодов состояния

- **1xx** (Информационный)
 - 102 Processing («идёт обработка»)
- **2xx** (Успех)
 - 200 OK («хорошо»)
- **3xx** (Перенаправления)
 - 301 Moved Permanently («перемещено навсегда»)
- **4xx** (Ошибка клиента)
 - 404 Not Found («не найдено»)
- **5xx** (Ошибка сервера)
 - 500 Internal Server Error («внутренняя ошибка сервера»)

Метод GET

- Способ посылки параметров сценарию, когда данные помещаются в командную строку URL, называется методом GET.
- **Пример:**
`http://example.com/hello.php?name=Vasya`

Метод POST

Метод запроса **POST** предназначен для запроса, при котором веб-сервер принимает данные, заключенные в тело сообщения, для хранения. Он часто используется для загрузки файла или представления заполненной веб-формы.

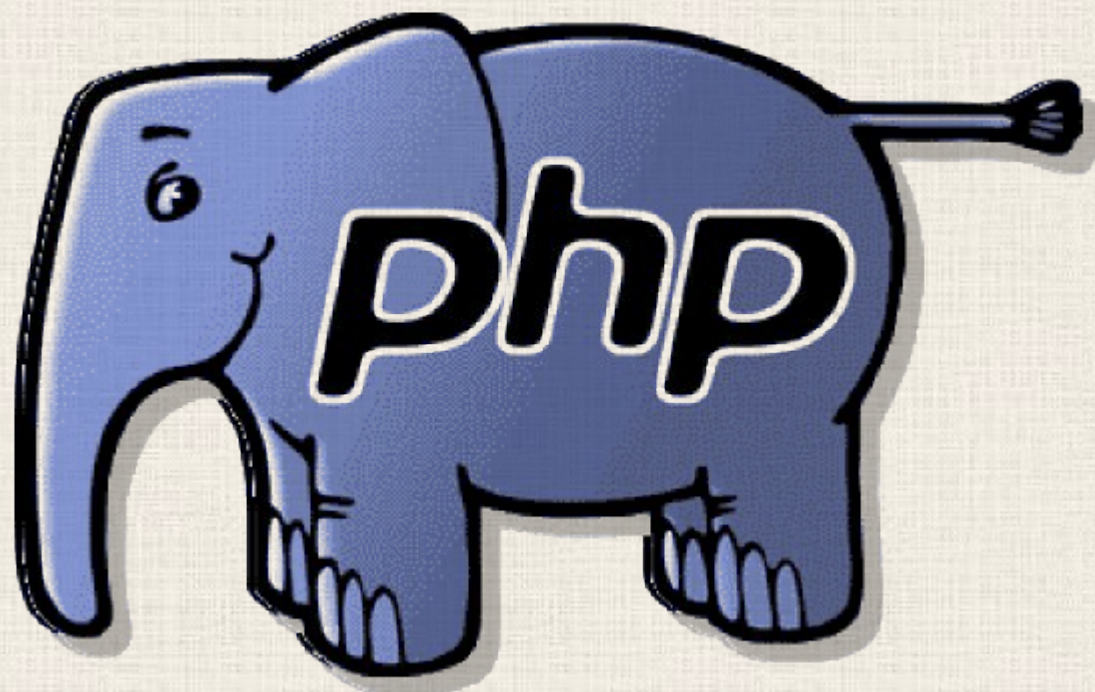
Заголовки

- **Content-Type:** text/plain; charset=utf-8
- **Server:** Apache/2.2.11 (Win32) PHP/5.3.0
- **Content-Language:** ru
- **Location:** http://example.com/page.html
- **User-Agent:** Mozilla/5.0 (Windows NT 6.1; WOW64)

CGI (Common Gateway Interface)

CGI — стандарт интерфейса, используемого для связи внешней программы с веб-сервером.





PHP (Hypertext Preprocessor)

PHP - скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений.

RНР

Создан Расмусом Лердорфом в 1995 г.



Основы синтаксиса PHP

- Теги PHP

- `<?php` (исполняемый код) `?>`

- `<?` (исполняемый код) `?>`

- Пример: `<? echo "test"; ?>`

- Разделение конструкций с помощью символа “;”

- Пример:

- `<? echo "Привет "; echo "Олег"; ?>`

Основы синтаксиса PHP

- **Комментарии (`/* */` , `//` , `#`)**

`/*` Мой

многострочный

комментарий `*/`

`//` Однострочный комментарий

`#` Однострочный комментарий

Пример программы

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Тестируем PHP</title>  
  </head>  
  <body>  
    <?php echo 'Hello, world!'; ?>  
  </body>  
</html>
```

Правила написания переменных

- Переменные в РНР обозначаются знаком доллара (\$), за которым должно следовать имя переменной.

Пример: \$name

Правила написания переменных

- **Имя переменной** после знака доллара должно начинаться с буквы или символа подчеркивания и далее может состоять из букв, цифр и символов подчеркивания в любом количестве.
- **Обратите внимание**, что имена переменных чувствительны к регистру букв.
`$name` и `$Name` - **разные переменные!**

ФУНКЦИИ

- `void echo (string $arg1 [, string $...]);`
- `int print (string $arg);`
- Функции `print` и `echo` - выводят строку

Оператор конкатенации

- Оператор конкатенации (`.`), который возвращает строку, представляющую собой соединение левого и правого аргумента.
- `$name = "Вася";`
- `$hello = "Привет";`
- `echo $hello . $name;`

:)