

# Интерактивное программирование веб-приложений

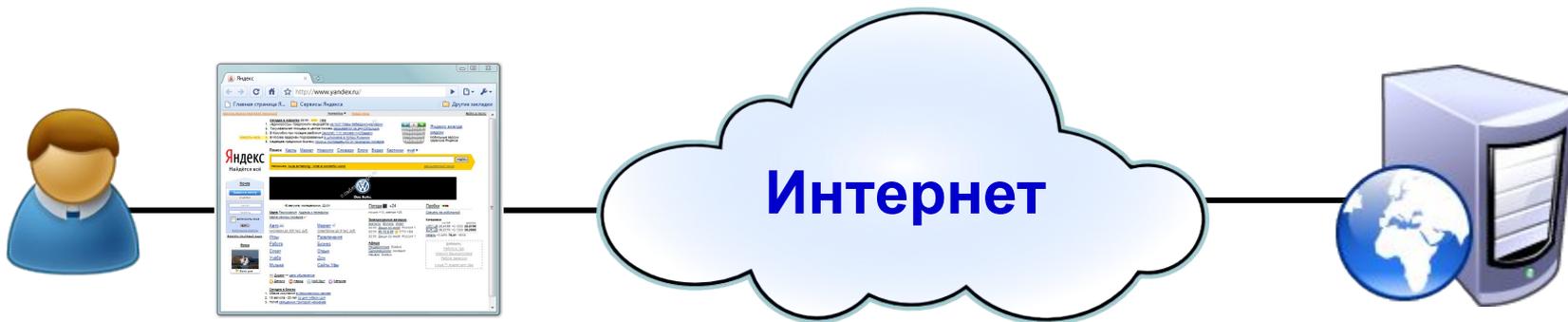
---

Преподаватель:  
Шварц Александр Юрьевич

# Принципы HTTP

`http://myserver.ru/news/main.php?year=2011&count=10`

HTTP-запрос



Ответ веб-сервера

(HTML-текст, изображение, CSS, JavaScript, SWF и т.п.)

# Запросы HTTP

---

- GET
  - Параметры отображаются в адресной строке  
URL?`user=17&page=news&id=134`
  - Используется для получения ресурса, без изменения
- POST
  - Используется для передачи пользовательских данных (например, отправки форм, загрузки на сервер файлов)

# Веб-сервер

---

**Веб-сервер** — это сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, обычно вместе с HTML-страницей, изображением, файлом, медиа-потокком или другими данными

**Примеры:** Apache, IIS, nginx, lighttpd



# Apache

---

- Кроссплатформенность
- Поддержка многих языков программирования
- Работа с СУБД, сессиями, аутентификацией, сообщениями об ошибках
- Поддержка IPv6
- Модульность

# PHP

---

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
- взаимодействие с большим количеством различных систем управления базами данных (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Интерфейс PDO);
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами, сокетами;
- обработка файлов, загружаемых на сервер.

# Примеры PHP-страниц

---

## Пример 1

```
<?php //открывающий тег
    echo "Hello, world!"; //вывод текста на страницу
?> //закрывающий тег
```

## Пример 2

```
Сейчас <?php echo date("Y"); ?> год
```

# Комментарии

---

КОД //однострочный комментарий

КОД

/\*

многострочный

комментарий

\*/

КОД

# Типы данных

---

## □ Скалярные типы данных

1. Двоичные данные (boolean) (true, false)
2. Целые числа (Integer) (1, -325, 5,...)
3. Числа с плавающей точкой (Float) (-0.001, 8.2, 436.4,...)
4. Строки (String) (“Hello, world!”, “Мама мыла раму”,...)

## □ Смешанные типы данных

1. Массивы (Array)
2. Объекты (Object)

## □ Специальные типы данных

1. Ресурсы (Resource)
2. Пустой тип (NULL)

# Переменные

---

## Пример 1

```
<?php
    $i = 7;
    echo $i; //7
    $s = "Google";
    echo $s; //Google
    $f = 8.9923;
    echo $f; //8.9923
    $b = true;
    echo $b; //1
    $i = $s;
    echo $i; //Google
?>
```

# Приведение типов

---

- (int), (integer) - приведение к целому числу
- (bool), (boolean) - приведение к булеву типу
- (float), (double), (real) - приведение к числу с плавающей точкой (float)
- (string) - приведение к строке
- (array) - приведение к массиву
- (object) - приведение к объекту

## Пример

```
$i = 10;    // $i это целое число
```

```
$b = (boolean) $i;    // $b это булев тип
```

# Преобразование в boolean

---

К значению false приводятся:

- сам булев **FALSE**
- целое 0 (ноль)
- число с плавающей точкой 0.0 (ноль)
- пустая строка и строка "0"
- массив с нулевыми элементами
- объект с нулевыми переменными-членами
- специальный тип NULL (включая неустановленные переменные)

Все остальные значения преобразуются к true!

# Преобразование строк в числа

---

```
<?php
```

```
$f = 1 + "10.5"; // $f это float(11.5)
$f = 1 + "-1.3e3"; // $f это float(-1299)
$f = 1 + "bob-1.3e3"; // $f это integer(1)
$f = 1 + "bob3"; // $f это integer(1)
$f = 1 + "10 o'clock"; // $f это integer(11)
$f = 4 + "10.2 kg"; //$f это float(14.2)
$f = "10.0 kg" + 1; //$f это float (11)
$f = "10.0 kg" + 1.0; //$f это float (11)
```

```
?>
```

# Операции со строками

---

```
<?php
    $i = "One";
    $b = "$i Two";
    echo "$b Three"; // One Two Three
    echo '$b Three'; // $b Three

    $s = "One" . "Two" . "Three"; // OneTwoThree
    $s .= "Four";
    echo $s; // OneTwoThreeFour
?>
```

# Массивы

---

```
$a = array(); // создание пустого массива
$b = array(1, 6, 43.56, 3, 2, 5, "Hello");
$a[] = 5; //добавление элемента в конец массива
$b[3] = 4; // замена 3-го элемента массива
```

```
$c = array(0=>34,
           3=>323525,
           4=>26,
           17=>23456);
$d = array("John"=>array(),
           "Mary"=>12,
           6=>"wd",
           1=>2);
```

# Работа с массивами

---

```
<?php
    $a = array(1, 2, 3, 4, 5);
    for($i = 0; $i < count($a); $i++)
    {
        echo $i . "=" . $a[$i] . "<br>";
    }

    foreach($a as $value)
    {
        echo $value . "<br>";
    }
    foreach($a as $key=>$value)
    {
        echo $key . "=" . $value . "<br>";
    }
?>
```

# ФУНКЦИИ

---

```
<?php
    function sum($x, $y)
    {
        return $x + $y;
    }
    function print_hello()
    {
        echo "Hello!";
    }
    function print_name($name = "Ivan")
    {
        echo $name;
    }
    $sum = sum(4, 7); //$sum == 11
    print_hello(); //Hello!
    print_name("Vasya"); //Vasya
    print_name(); //Ivan
?>
```

# Переменные переменных

---

```
<?php
```

```
    $animal = "cat";
```

```
    $cat = 4;
```

```
    $spider = 8;
```

```
    echo $animal; //cat
```

```
    echo $$animal; //4
```

```
    $animal = "spider";
```

```
    echo $$animal; //8
```

```
    $$animal = 7;
```

```
    echo $spider; //7
```

```
?>
```

# Глобальные массивы

---

- `$_GET`
- `$_POST`
- `$_SESSION`
- `$_COOKIE`
- `$_SERVER`
- `$_ENV`
- `$_REQUEST`
- `$_FILES`

# Массив `$_SERVER`

---

`$_SERVER["OS"]` – операционная система

`$_SERVER["SERVER_SOFTWARE"]` – веб-сервер

`$_SERVER["SERVER_NAME"]` – имя сервера

`$_SERVER["SERVER_ADDR"]` – адрес сервера

`$_SERVER["SERVER_PORT"]` – порт сервера

`$_SERVER["REMOTE_ADDR"]` – адрес клиента

`$_SERVER["DOCUMENT_ROOT"]` – путь к документам на сервере

`$_SERVER["SCRIPT_FILENAME"]` – полный путь к текущему скрипту

`$_SERVER["PHP_SELF"]` – имя текущего скрипта

# Пример работы с формами

---

form.php

```
<html>
  <body>
    <form method="POST" action="result.php">
      <input type="text" name="username"><br>
      <input type="text" name="password"><br>
      <input type="submit">
    </form>
  </body>
</html>
```

# Пример работы с формами - 2

---

## result.php

```
<?php
    if(isset($_POST["username"]))
    {
        echo "Hello, " . $_POST["username"] . "<br>";
    }
    else echo "Hello, somebody!<br>";

    if(isset($_POST["password"]))
    {
        if($_POST["password"] == "123456") echo "Logged in!";
    }
    else echo "Wrong password!";

?>
```

# Работа с сессиями

---

```
<?php
    session_start(); //начинаем работу сессий
    if(isset($_SESSION["enter_time"]))
    {
        $dif = time() - $_SESSION["enter_time"];
        echo "You entered " . $dif . " s ago";
    }
    else
    {
        $_SESSION["enter_time"] = time();
        echo "Entered now";
    }
?>
```

# Работа с файлами

---

`fopen ($filename , $flag) ;`

- **r** – открытие файла только для чтения.
- **r+** - открытие файла одновременно на чтение и запись.
- **w** – создание нового пустого файла. Если на момент вызова уже существует такой файл, то он уничтожается.
- **w+** - аналогичен **r+**, только если на момент вызова файл существует, его содержимое удаляется.
- **a** – открывает существующий файл в режиме записи, при этом указатель сдвигается на последний байт файла (на конец файла).
- **a+** - открывает файл в режиме чтения и записи при этом указатель сдвигается на последний байт файла (на конец файла). Содержимое файла не удаляется.

# Работа с файлами - 2

---

## Дозапись в конец файла

```
<?php
    $fp = fopen("counter.txt", "a"); // Открываем
    файл в режиме записи
    $mytext = "Text\r\n"; // Исходная строка
    $test = fwrite($fp, $mytext); // Запись в файл
    if ($test) echo "Success!";
    else echo "Error";
    fclose($fp); //Закрытие файла
?>
```

# Работа с файлами - 3

---

```
<?php
```

```
    // Считывание файла в массив $file_array  
    $file_array = file("counter.txt");
```

```
    // Построчный вывод содержимого файла  
    foreach($file_array as $s)  
    {  
        echo $s . "<br>";  
    }
```

```
?>
```

# Работа с файлами - 4

---

- `file_exists($name)` – проверка существования файла
- `filesize($name)` – вычисление размера файла в байтах
- `file_put_contents($name, $text)` – запись данных в файл
- `file_get_contents($name)` – получение данных из файла

# Подключение скриптов

---

- **include, require** – загрузка и выполнение скрипта
- **include\_once, require\_once** – загрузка и выполнение скрипта, без возможности повторной загрузки и выполнения

**vars.php**

```
<?php
$color = 'green';
$fruit = 'apple';
?>
```

**test.php**

```
<?php
echo "A $color $fruit";// A
include 'vars.php';
echo "A $color $fruit";//A green apple
?>
```

# Работа с СУБД MySQL

---

Соединение с СУБД:

```
mysql_connect ($hostname, $user, $password) ;
```

Соединение с базой данных

**\$hostname** – имя хоста, например, 192.168.2.1:3360, localhost или 81.12.0.54;

**\$user** – имя пользователя в MySQL;

**\$password** – пароль пользователя в MySQL.

Возвращается целочисленный идентификатор соединения.

Закрытие соединения - **mysql\_close ()**

Постоянное соединение с СУБД:

```
mysql_pconnect ($hostname, $user, $password) ;
```

# Пример

---

```
// Попытка установить соединение с MySQL
if (!mysql_connect("localhost", "root", "mypass"))
{
    echo "Ошибка подключения";
    exit;
}
// Соединились, теперь выбираем базу данных
mysql_select_db("shop");
```

# Обработка ошибок

---

Получение номера последней ошибки

```
mysql_errno($id)
```

Получение сообщения о последней ошибке

```
mysql_error($id)
```

# Обработка результатов запроса

---

```
//Получение результата выполнения запроса
$q = mysql_query("SELECT * FROM mytable");
//количество возвращенных строк
$count = mysql_num_rows($q);
for ($c = 0; $c < $count; $c++)
{
    echo "Строка №" . $c . "<br>";
    //получение одной строки из результата
    $f = mysql_fetch_array($q);
    foreach($f as $k=>$v)
    {
        echo $k . "=" . $v . "<br>";
    }
}

mysql_free_result($q); //освобождение памяти
```

# Полезные ссылки

---

- <http://www.php.net>
- <http://www.php.su>
- <http://www.php.ru>
- <http://www.w3schools.com/php/default.asp>
- РНР 5 В Подлиннике - Д. Котеров, А. Костарев
- РНР и MySQL. Библия программиста – С. Суэринг, Т. Конверс, Д. Парк