



# Web-технологии

Применение языка PHP

- Настройка nginx для работы с php
- Знакомство с PHP
  - область применения
  - основные элементы языка (переменные, условные операторы, циклы, строки, массивы, функции, элементы ООП)
  - работа с файлами (в т.ч. xml)
  - обработка запросов с клиента (get, post)
  - обработка исключительных ситуаций и ошибок
- Критика языка
- Пара слов про CMS

# PHP: Hypertext Preprocessor

- Набор встроенных средств для разработки веб-приложений:
  - автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы
  - взаимодействие с большим количеством различных систем управления базами данных 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
  - работа с XForms
- Что может PHP?
  - PHP может создавать страницы с динамическим содержимым
  - PHP может создавать, открывать, читать, писать, удалять и закрывать файлы на сервере
  - PHP может собирать данные из формы
  - PHP может отправлять и получать cookies
  - PHP может добавлять, удалять, изменять данные в БД
  - PHP может использоваться для контроля доступа пользователем
  - PHP может шифровать данные

# PHP – скриптовый язык

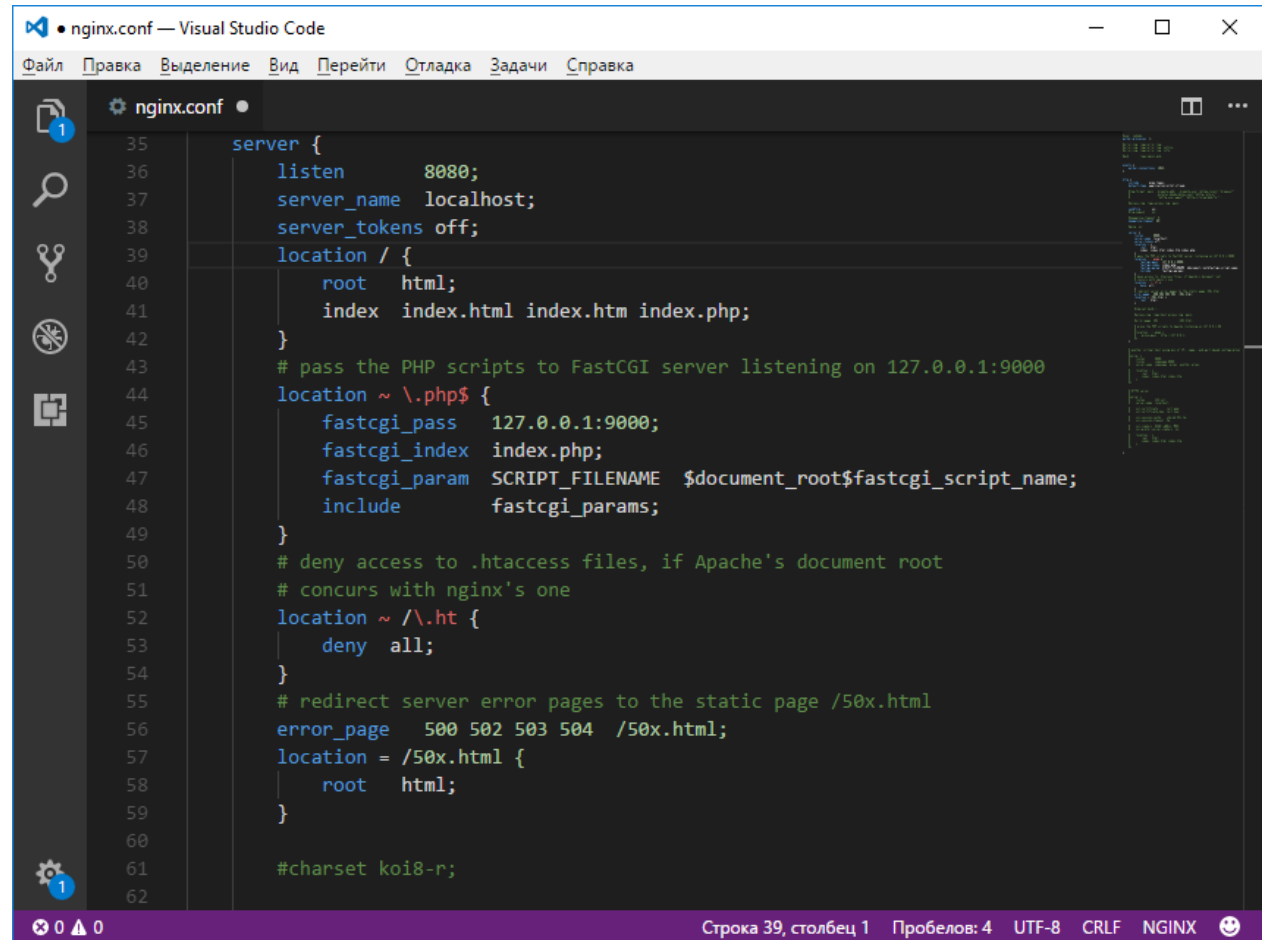
4

The screenshot shows the official PHP website. The browser's address bar displays 'www.php.net'. The website's navigation bar includes links for 'Downloads', 'Documentation', 'Get Involved', and 'Help', along with the PHP logo and a search bar. The main content area features a dark background with white text. On the left, it states 'PHP is a popular general-purpose scripting language that is especially suited to web development.' and 'Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.' On the right, there is a 'Download' section listing versions 8.0.12, 7.4.25, and 7.3.31, each with links for 'Release Notes' and 'Upgrading'. Below this, there are sections for 'Upcoming conferences' (listing 'PHP Russia 2022'), 'User Group Events', 'Special Thanks', and 'Social media' (listing '@official\_php'). A large white box on the left side of the page contains a news item titled 'PHP 7.4.25 Released! »' dated '22 Oct 2021'. The text of the news item states: 'The PHP development team announces the immediate availability of PHP 7.4.25. This is a security release. All PHP 7.4 users are encouraged to upgrade to this version. For source downloads of PHP 7.4.25 please visit our [downloads page](#), Windows source and binaries can be found on [windows.php.net/download/](#). The list of changes is recorded in the [ChangeLog](#).'

<https://www.php.net/>

# Настройка nginx – windows / nginx.conf

5



```
35  server {
36      listen      8080;
37      server_name  localhost;
38      server_tokens off;
39      location / {
40          root      html;
41          index      index.html index.htm index.php;
42      }
43      # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
44      location ~ /\.php$ {
45          fastcgi_pass    127.0.0.1:9000;
46          fastcgi_index    index.php;
47          fastcgi_param    SCRIPT_FILENAME    $document_root$fastcgi_script_name;
48          include           fastcgi_params;
49      }
50      # deny access to .htaccess files, if Apache's document root
51      # concurs with nginx's one
52      location ~ /\.ht {
53          deny    all;
54      }
55      # redirect server error pages to the static page /50x.html
56      error_page   500 502 503 504 /50x.html;
57      location = /50x.html {
58          root      html;
59      }
60
61      #charset koi8-r;
62  }
```

## Настройка nginx с PHP под Windows:

<https://www.easycoding.org/2011/05/09/ustanovka-i-nastrojka-nginx-php-pod-windows.html>

Распаковать **php** в папку **nginx**

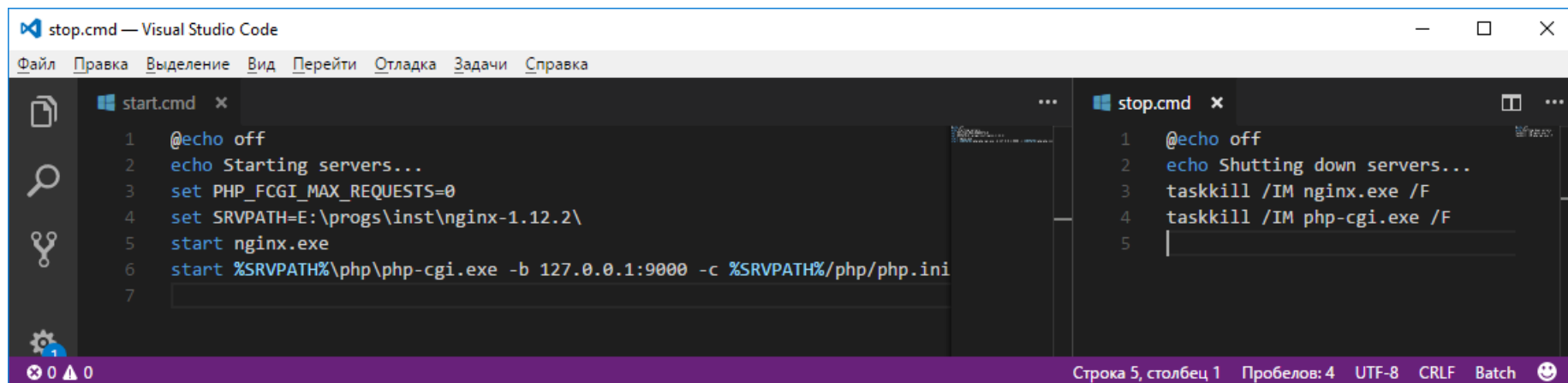
# Настройка nginx – windows / start-stop

## start.cmd

```
@echo off
echo Starting servers...
set PHP_FCGI_MAX_REQUESTS=0
set SRVPATH=E:\progs\inst\nginx-1.12.2\
start nginx.exe
start %SRVPATH%\php\php-cgi.exe -b 127.0.0.1:9000 -c %SRVPATH%\php\php.ini
```

## stop.cmd

```
@echo off
echo Shutting down servers...
taskkill /IM nginx.exe /F
taskkill /IM php-cgi.exe /F
```



The screenshot shows the Visual Studio Code editor with two files open: start.cmd and stop.cmd. The start.cmd file contains commands to start nginx and php-cgi. The stop.cmd file contains commands to shut down nginx and php-cgi. The status bar at the bottom indicates the current cursor position is at line 5, column 1, with 4 spaces, UTF-8 encoding, CRLF line endings, and Batch file type.

```
stop.cmd — Visual Studio Code
Файл  Правка  Выделение  Вид  Перейти  Отладка  Задачи  Справка

start.cmd x
1  @echo off
2  echo Starting servers...
3  set PHP_FCGI_MAX_REQUESTS=0
4  set SRVPATH=E:\progs\inst\nginx-1.12.2\
5  start nginx.exe
6  start %SRVPATH%\php\php-cgi.exe -b 127.0.0.1:9000 -c %SRVPATH%\php\php.ini
7

stop.cmd x
1  @echo off
2  echo Shutting down servers...
3  taskkill /IM nginx.exe /F
4  taskkill /IM php-cgi.exe /F
5
```

Строка 5, столбец 1 Пробелов: 4 UTF-8 CRLF Batch

# Настройка nginx – Ubuntu / install

7

- Установка программ
  - **sudo apt-get install nginx nginx-extras**
  - **sudo apt-get install php-cli php-common php-mysql php-gd php-fpm php-cgi php-fpm php-pear php-mcrypt**
- Остановка служб
  - **sudo service nginx stop**
  - **sudo service php7.0-fpm stop**
- Настройка PHP
  - **sudo nano /etc/php/7.0/fpm/php.ini**
    - Защита от запросов вида «/image.gif/foo.php»
    - `;cgi.fix_pathinfo = 1`
    - `cgi.fix_pathinfo = 0`
  - **sudo nano /etc/php/7.0/fpm/pool.d/www.conf**
    - Раскомментировать (ограничение выполнения файлов по расширению)
      - `security.limit_extensions = .php .php3 .php4 .php5 .php7`
    - Исправить (файл для связи «Nginx» с «PHP-FPM» - сокет)
      - `listen = /var/run/php/php7.0-fpm.sock`
    - Добавить
      - `listen.owner = www-data`
      - `listen.group = www-data`
      - `listen.mode = 0660`
  - **sudo service php7.0-fpm restart**
- Проверить права
  - **srw-rw---- 1 www-data www-data 0 янв 19 12:46 php7.0-fpm.sock**

# Настройка nginx – Ubuntu / config

- **sudo nano /etc/nginx/sites-available/default**

- Раскомментировать

```
location ~ \.php$ {  
    include snippets/fastcgi-php.conf;  
    # With php7.0-cgi alone:  
    #fastcgi_pass 127.0.0.1:9000;  
    # With php7.0-fpm:  
    fastcgi_pass unix:/run/php/php7.0-fpm.sock;  
}
```

- **sudo service nginx start**

- Дать права доступа к папке

- **sudo chmod -R 777 /var/www/html**

- **nano /var/www/html/info.php**

- Создать файл

```
<?php  
echo "Hello world";  
phpinfo();  
?>
```

- В браузере

- **http://localhost/info.php**



# Hello world

9

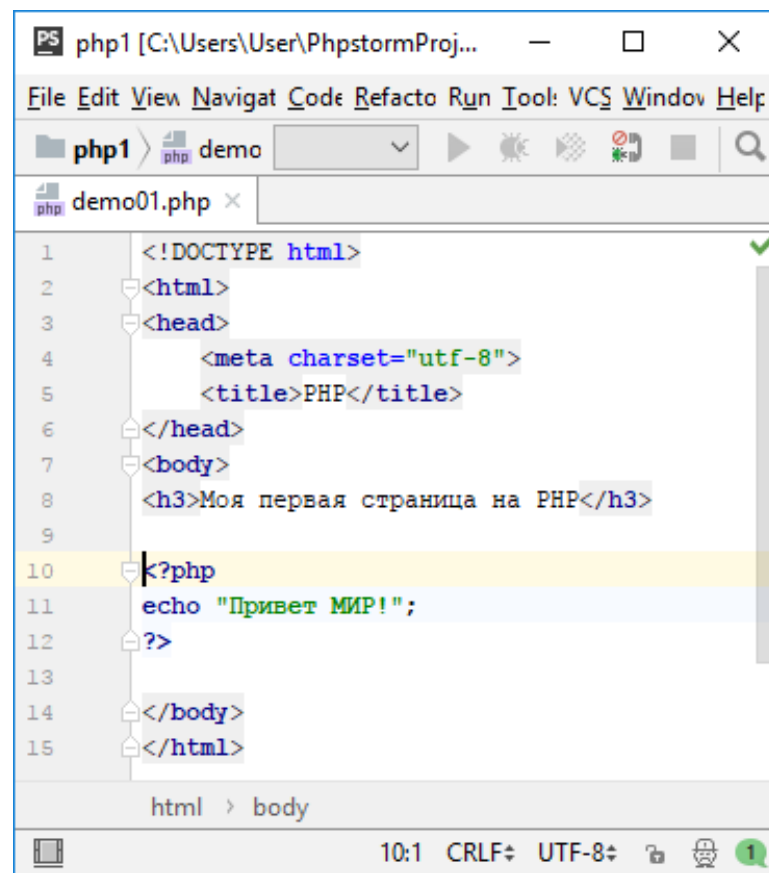
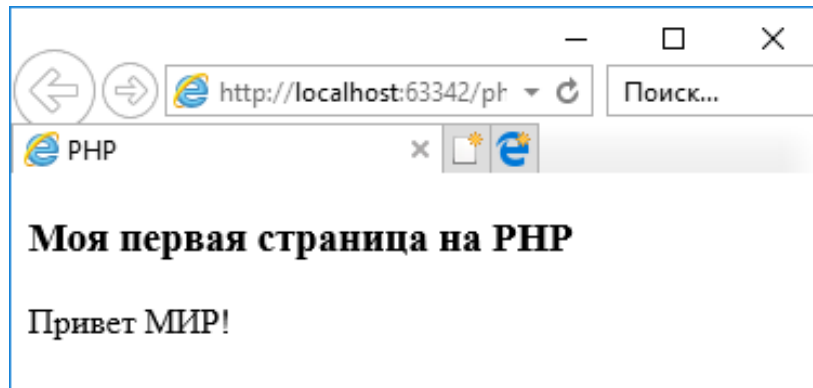
## demo01.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>PHP</title>
</head>
<body>
<h3>Моя первая страница на PHP</h3>

<?php
echo "Привет МИР!";
?>

</body>
</html>
```

<?php код пишем здесь ?>

A screenshot of a code editor window titled 'php1 [C:\Users\User\PhpstormProj...'. The editor shows the contents of 'demo01.php' with line numbers 1 through 15. The code is: 1 <!DOCTYPE html>, 2 <html>, 3 <head>, 4 <meta charset="utf-8">, 5 <title>PHP</title>, 6 </head>, 7 <body>, 8 <h3>Моя первая страница на PHP</h3>, 9, 10 <?php, 11 echo "Привет МИР!";, 12 ?>, 13, 14 </body>, 15 </html>. The status bar at the bottom shows '10:1 CRLF UTF-8'.

## demo02.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>PHP</title>
</head>
<body>
<h3>Моя первая страница на PHP</h3>

<?php
// Однострочный комментарий
# Однострочный комментарий
/*
 * Многострочный
 * комментарий
 */
echo "Привет МИР!";
?>

</body>
</html>
```

- Однострочные комментарии
  - //
  - #
- Многострочные комментарии
  - /\* \*/

# Ключевые слова case insensitive

11

## demo03.php

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>PHP</title>
</head>
<body>
<h3>Моя первая страница на PHP</h3>

<?php
ЕCHO "Привет МИР!";
ЕсНо "Привет МИР!";
echo "Привет МИР!";
?>

</body>
</html>
```

- Имена переменных учитывают регистр букв (case sensitive)
  - \$color
  - \$Color
  - \$COLOR
  - разные переменные

# Переменные в PHP (1)

12

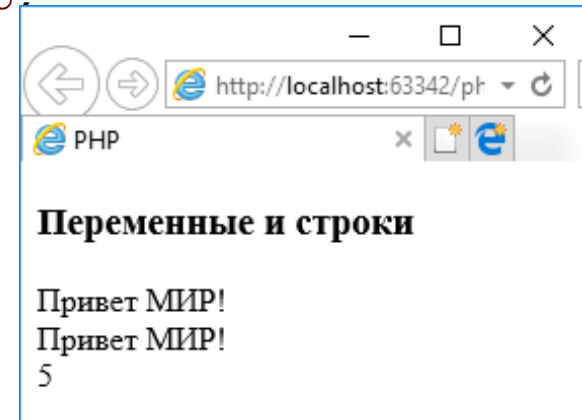
- Переменные начинаются со знака \$, за которым следует имя переменной
- Имя переменной должно начинаться с буквы или знака подчёркивание
- Имя переменной не может начинаться с цифры
- Переменная может состоять только из латинских букв (в любом регистре), цифр и знака подчёркивания
- Имя переменной учитывает регистр букв

## demo04.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>PHP</title>
</head>
<body>
<h3>Переменные и строки</h3>

<?php
$txt = "МИР";
ЕCHO "Привет $txt!<br>";
ЕCHO "Привет " . $txt . "!<br>";
$a = 2;
$b = 3;
echo $a + $b;
?>

</body>
</html>
```



# Переменные в PHP (2) – область видимости

## demo05.php

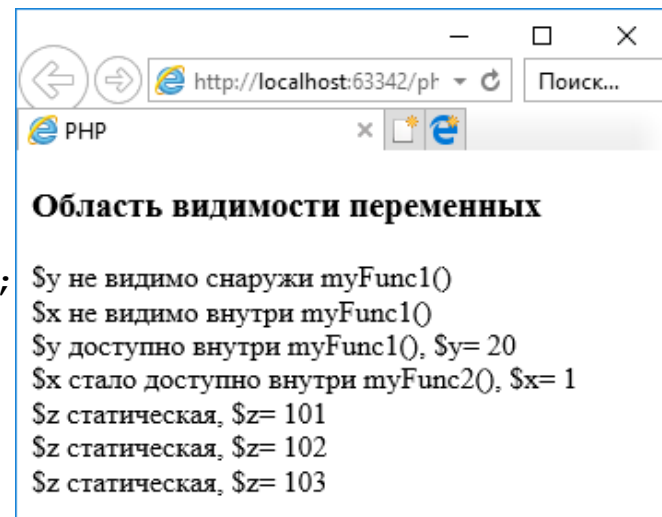
<h3>Область видимости переменных</h3>

```
<?php
$x = 1;
function myFunc1 () {
    $y = 20;
    print "\$x не видимо внутри myFunc1()<br>";
    print "\$y доступно внутри myFunc1(), \$y= $y<br>";
}
print "\$y не видимо снаружи myFunc1()<br>";

function myFunc2 () {
    global $x;
    echo "\$x стало доступно внутри myFunc2(), \$x= $x<br>";
}

function myFunc3 () {
    static $z = 100;
    $z++;
    print "\$z статическая, \$z= $z<br>";
}

myFunc1();
myFunc2();
myFunc3();
myFunc3();
myFunc3();
?>
```



**echo === print**

# Типы переменных в PHP

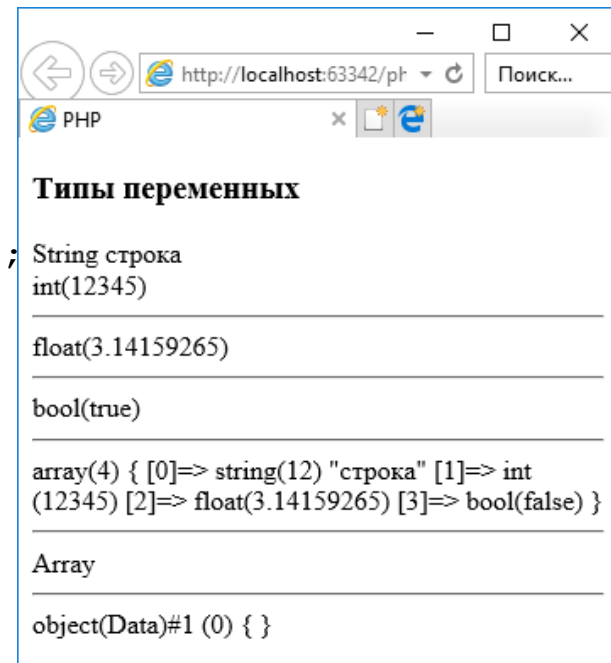
14

- String – строка
- Integer – целое
- Float (double) – вещественное
- Boolean – логический
- Array – массив
- Object – объект
- NULL
- Resource – ресурс

## demo06.php

```
<?php
$str = "строка";
echo "String $str<br>";
$int = 12345;
var_dump($int);
print "<hr>";
$f1 = 3.14159265;
var_dump($f1);
print "<hr>";
$bool = true;
var_dump($bool);
print "<hr>";
$arr = array($str, $int, $f1, !$bool);
var_dump($arr);
print "<hr>$arr";

print "<hr>";
class Data {
    function hw() {
        echo "Привет МИР";
    }
}
$ex = new Data();
var_dump($ex);
?>
```



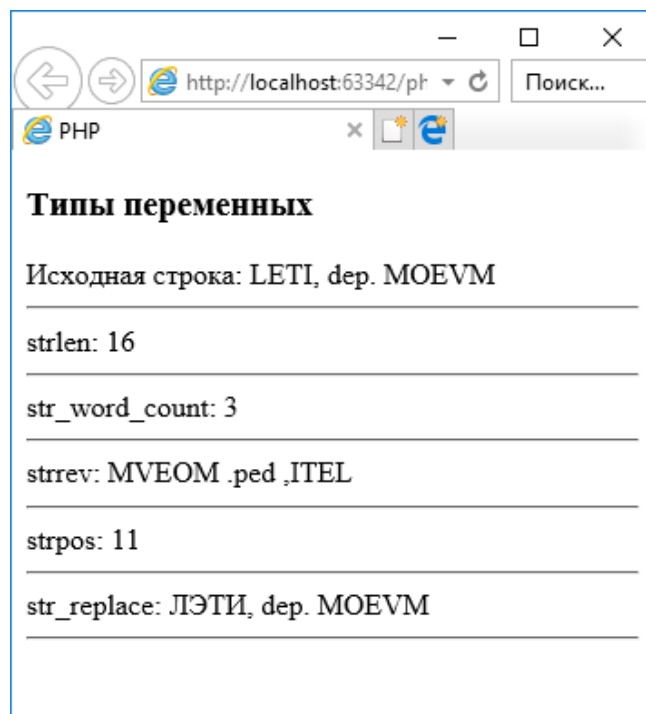
# Строки

15

## demo07.php

```
<?php
$str = "LETI, dep. MOEVM";
echo "Исходная строка: $str<hr>";
echo "strlen: " . strlen($str) . "<hr>";
echo "str_word_count: " . str_word_count($str) . "<hr>";
echo "strrev: " . strrev($str) . "<hr>";
echo "strpos: " . strpos($str, "MO") . "<hr>";
echo "str_replace: " . str_replace("LETI", "ЛЭТИ", $str) . "<hr>";

?>
```



# Константы

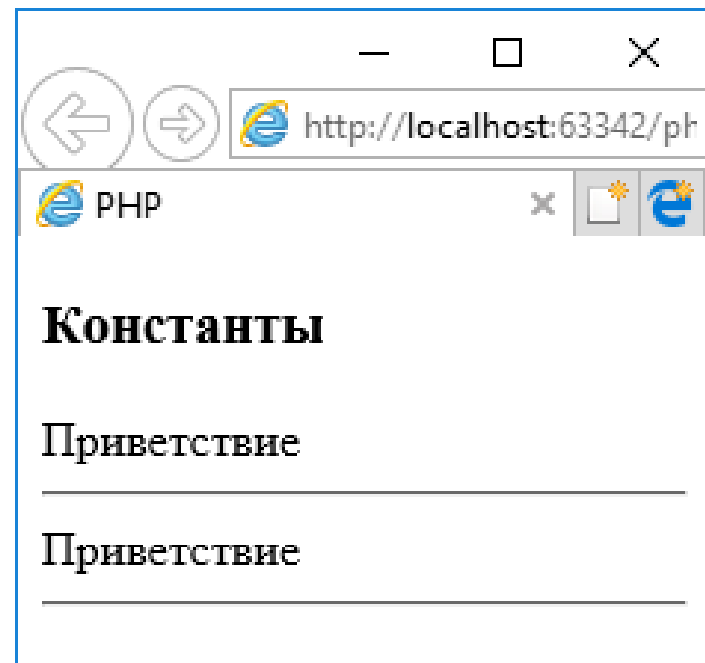
16

## demo08.php

```
<?php
define("GREETING", "Приветствие<hr>");
//define("GREETING", "Приветствие", true);
echo GREETING;
function myFunc() {
    echo GREETING; // Глобальные
}
myFunc()
?>
```

### Синтаксис:

define(name, value, case-insensitive)





## Арифметические

- +, -, \*, /, %, \*\*

## Присваивания

- =, +=, -=, /=, \*=, %=

## Сравнение

- ==, ===, !=, <>, !==, >, <, >=, <=

## Увеличение, уменьшение

- ++, --

## Логические

- and, or, xor, &&, ||, !

## Строковые

- ., .=

## Операции с массивами

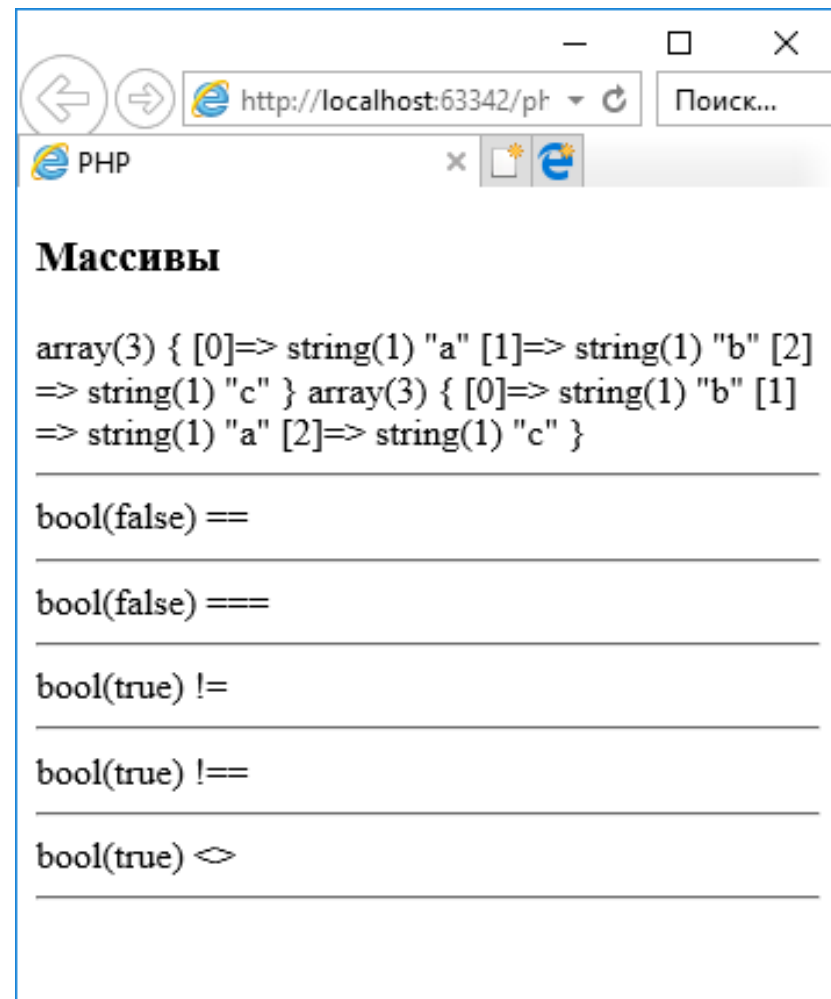
- +, ==, ===, !=, <>, !==

# Операции с массивами

18

## demo09.php

```
<?php
$x = array("a", "b", "c");
$y = array("b", "a", "c");
var_dump($x);
var_dump($y);
echo "<hr>";
var_dump($x == $y);
echo "==<hr>";
var_dump($x === $y);
echo "===<hr>";
var_dump($x != $y);
echo "!=<hr>";
var_dump($x !== $y);
echo "!==<hr>";
var_dump($x <> $y);
echo "<><hr>";
?>
```



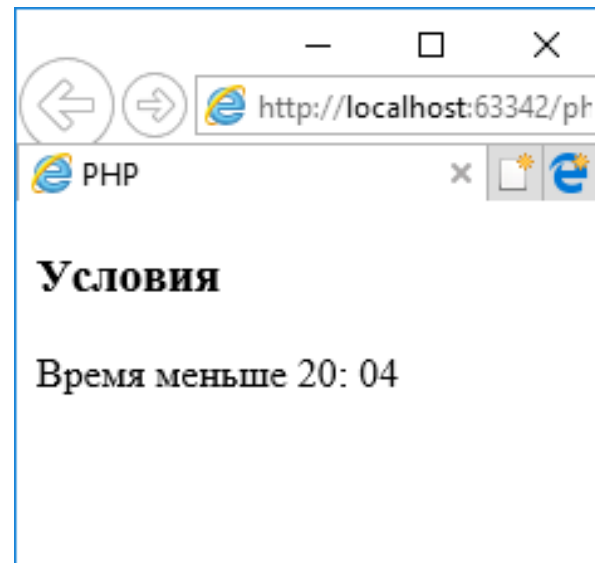
# Условные операторы

## demo10.php

```
<?php
$d = date("s");
if($d < 20)
    echo "Время меньше 20: $d";
elseif($d < 40)
    echo "Время меньше 40: $d";
else
    echo "Время: $d";
?>
```

### Синтаксис:

```
if (условие 1) {
    код 1
} elseif (условие 2) {
    код 2
} else {
    код 3
}
```

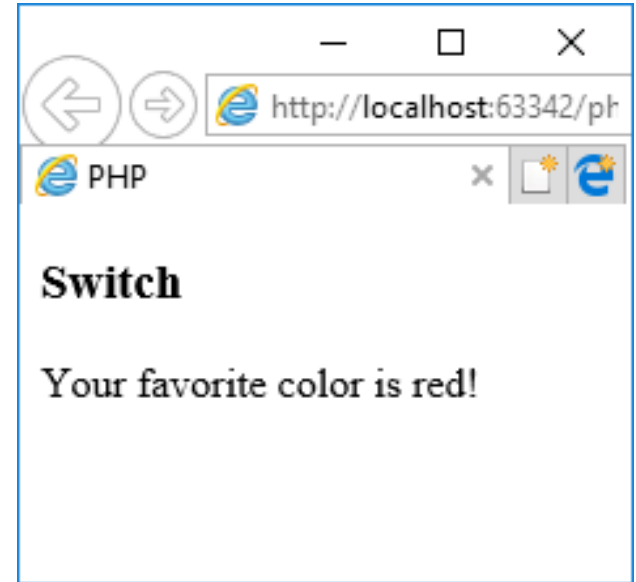


# Switch

## demo11.php

```
$favcolor = "red";

switch ($favcolor) {
    case "red":
        echo "Your favorite color is red!";
        break;
    case "blue":
        echo "Your favorite color is blue!";
        break;
    case "green":
        echo "Your favorite color is green!";
        break;
    default:
        echo "Your favorite color is neither red, blue, nor green!";
}
```

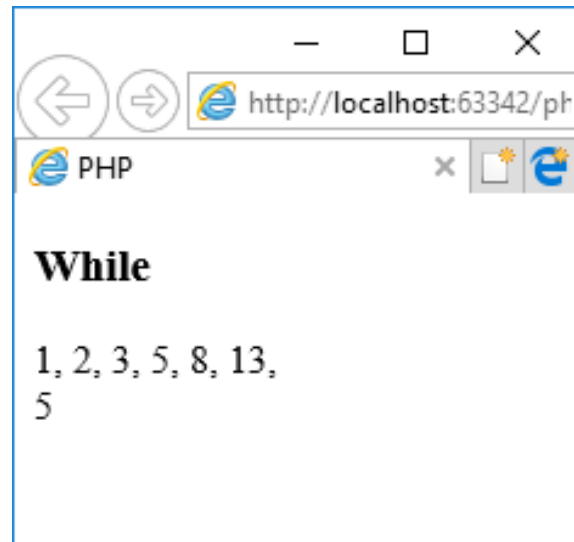


# While, for

21

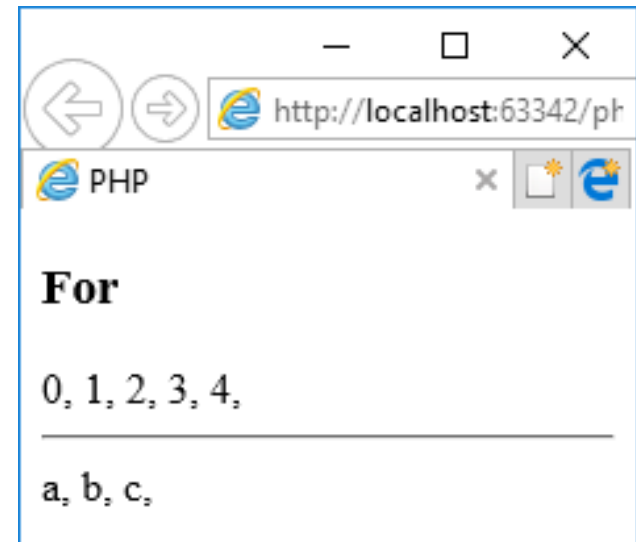
## demo12.php

```
<?php
$x1 = 1;
$x2 = 1;
while ($x2 < 15) {
    $x3 = $x1 + $x2;
    $x1 = $x2;
    $x2 = $x3;
    echo "$x1, ";
}
$y = 0;
do {
    $y++;
} while ($y < 5);
echo "<br>$y";
?>
```



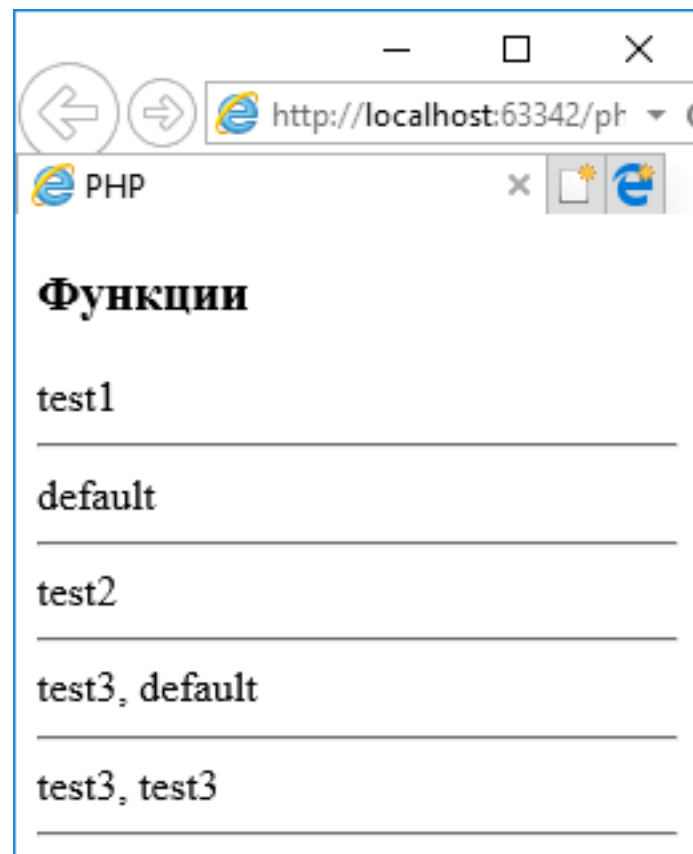
## demo13.php

```
<?php
for ($i = 0; $i < 5; $i++)
    echo "$i, ";
echo "<hr>";
$arr = array("a", "b", "c");
foreach ($arr as $x)
    echo "$x, ";
?>
```



## demo14.php

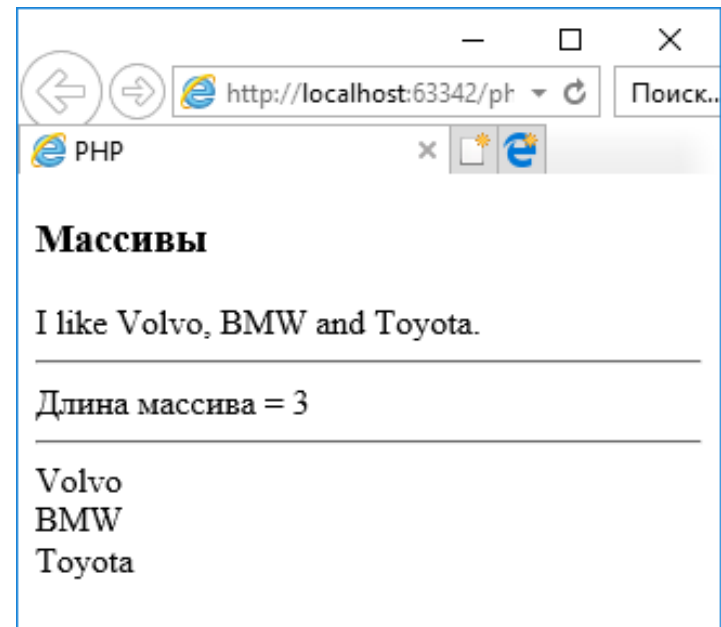
```
<?php
function myFunc1 ($x) {
    echo "$x<hr>";
}
function myFunc2 ($x = "default") {
    echo "$x<hr>";
}
function myFunc3 ($x, $y = "default") {
    return "$x, $y<hr>";
}
myFunc1 ("test1");
myFunc2 ();
myFunc2 ("test2");
echo myFunc3 ("test3");
echo myFunc3 ("test3", "test3");
?>
```



## demo15.php

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] .
      " and " . $cars[2] . ".";
echo "<hr>Длина массива = " . count($cars);
echo "<hr>";
$arlength = count($cars);

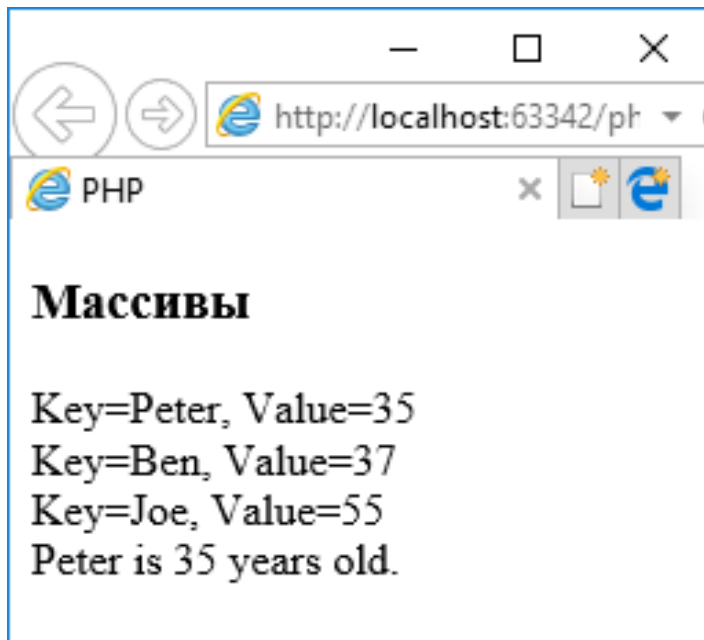
for($x = 0; $x < $arlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```



# «Ассоциативные» массивы

## demo16.php

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
$age['Joe'] = "55";
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
echo "Peter is " . $age['Peter'] . " years old.";
?>
```



**ключ => значение**

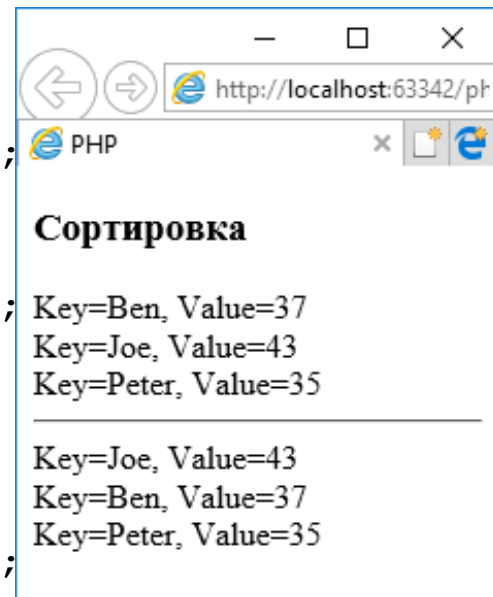


# Сортировка массива

25

## demo17.php

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
ksort($age);
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value . "<br>";
}
echo "<hr>";
arsort($age);
foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value . "<br>";
}
?>
```



- sort() – сортировка по возрастанию
- rsort() – сортировка по убыванию
- asort() – сортировка по возрастанию по значению
- ksort() – сортировка по возрастанию по ключу
- arsort() – сортировка по убыванию по значению
- krsort() – сортировка по убыванию по ключу

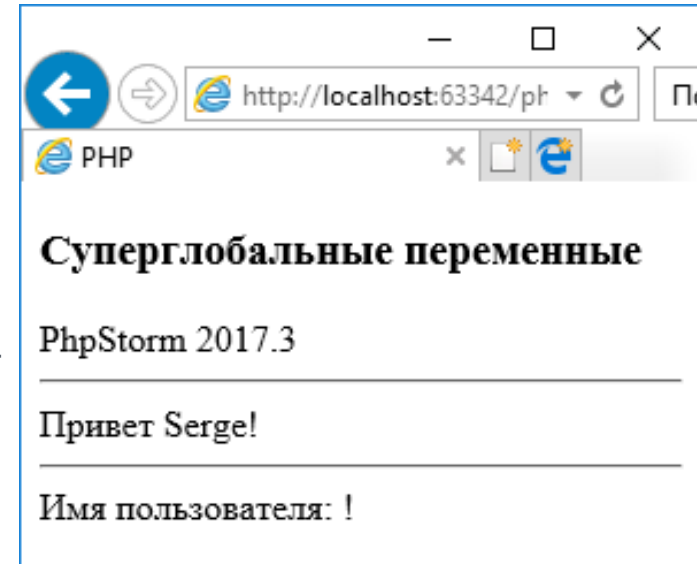
# Суперглобальные переменные (1)

26

## demo18.php

```
<?php
echo $_SERVER['SERVER_NAME'];
echo "<hr>";
echo 'Привет ' . htmlspecialchars($_GET["name"]) . '!';
echo "<hr>";
echo 'Имя пользователя: ' . $_ENV["USER"] . '!';
?>
```

- **\$GLOBALS** - Ссылки на все переменные глобальной области видимости
- **\$\_SERVER** - Информация о сервере и среде исполнения
- **\$\_GET** - GET-переменные HTTP, переданные скрипту через URL
- **\$\_POST** - POST-переменные HTTP, переданные скрипту через HTTP метод POST
- **\$\_FILES** - Переменные файлов, загруженных по HTTP
- **\$\_COOKIE** - HTTP Куки
- **\$\_SESSION** - Переменные сессии
- **\$\_REQUEST** - Переменные HTTP-запроса
- **\$\_ENV** - Переменные окружения



# Суперглобальные переменные (2)

27

demo19.php

<h3>Суперглобальные переменные</h3>

```
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
```

```
    Имя: <input type="text" name="fname">
```

```
    <input type="submit">
```

```
</form>
```

```
<?php
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    // Получить данные из input
```

```
    $name = $_REQUEST['fname'];
```

```
    if (empty($name)) {
```

```
        echo "Имя не введено";
```

```
    } else {
```

```
        echo $name;
```

```
    }
```

```
}
```

```
?>
```

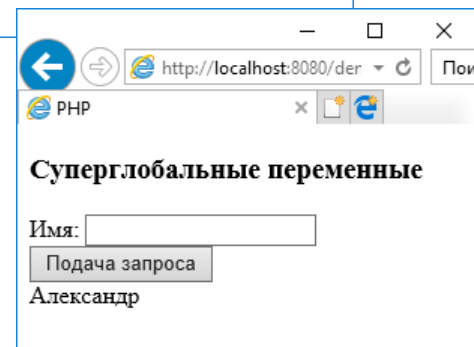
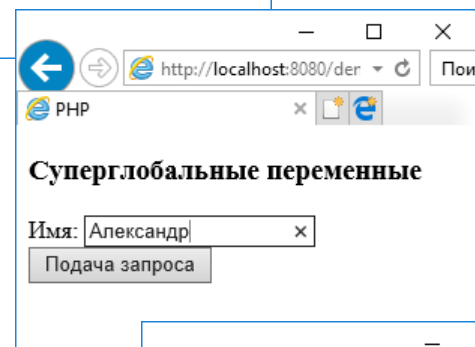
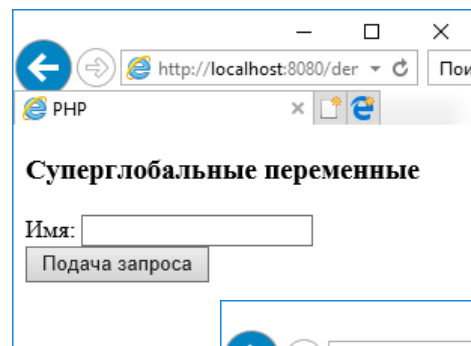
Защита от подмены переменной

PHP\_SELF

```
<form method="post" action="<?php
```

```
    echo htmlspecialchars($_SERVER["PHP_SELF"]);
```

```
?>">
```



# Метод GET

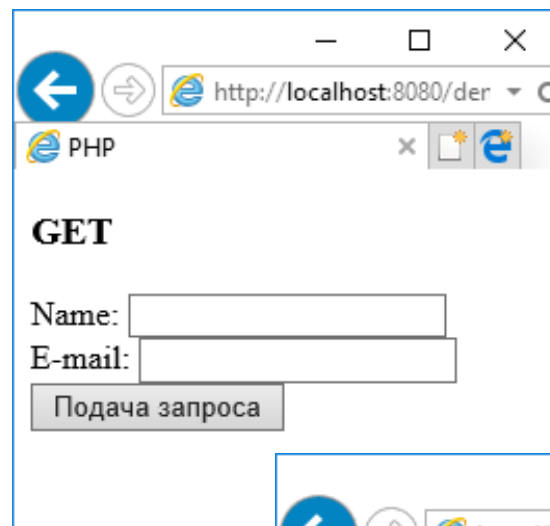
28

## demo20.html

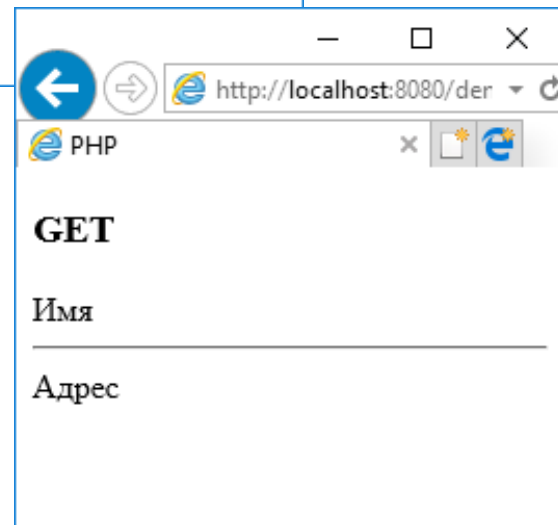
```
<form action="demo20.php" method="get">
  Name: <input type="text" name="name"><br>
  E-mail: <input type="text" name="email"><br>
  <input type="submit">
</form>
```

## demo20.php

```
<?php
if ($_GET["name"] != NULL) {
    $name = $_GET['name'];
    $email = $_GET['email'];
    if (empty($name)) {
        echo "Имя не введено";
    } else {
        echo $name;
    }
}
echo "<hr>";
if (empty($email)) {
    echo "Email не введен";
} else {
    echo $email;
}
?>
```



A screenshot of a web browser window. The address bar shows 'http://localhost:8080/der'. The page title is 'PHP'. The main content area displays the word 'GET' in bold. Below it, there are two text input fields labeled 'Name:' and 'E-mail:'. At the bottom, there is a button labeled 'Подача запроса' (Submit request).



A screenshot of a web browser window. The address bar shows 'http://localhost:8080/der'. The page title is 'PHP'. The main content area displays the word 'GET' in bold. Below it, the text 'Имя' (Name) is shown above a horizontal line, and 'Адрес' (Address) is shown below another horizontal line.

# Валидация

29

demo21.php

```
<?php
$nameErr = $genderErr = "";
$name = $gender = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"]))
        $nameErr = "Name is required";
    else
        $name = test_input($_POST["name"]);
    if (empty($_POST["gender"]))
        $genderErr = "Gender is required";
    else
        $gender = test_input($_POST["gender"]);
}
function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}
?>
```

## PHP валидация

\* обязательные поля.

Имя:  \*

Пол: ☐ Женский ☐ Мужской \*

Вы ввели:

## PHP валидация

\* обязательные поля.

Имя:  \* Name is  
required

Пол: ☐ Женский ☐ Мужской \*

Вы ввели:

male

```
<h3>PHP валидация</h3>
<p><span class="error">* обязательные поля.</span></p>
<form method="post" action="<?php
    echo htmlspecialchars($_SERVER["PHP_SELF"]);
?>">
Имя: <input type="text" name="name">
<span class="error">* <?php echo $nameErr; ?></span> <br>
Пол:
<input type="radio" name="gender" value="female">Женский
<input type="radio" name="gender" value="male">Мужской
<span class="error">* <?php echo $genderErr; ?></span> <br>
<input type="submit" name="submit" value="Submit">
</form>

<?php
echo "<h3>Вы ввели:</h3>";
echo $name;
echo "<br>";
echo $gender;
?>
```

# Регулярные выражения

## demo22.php

```
<h3>Регулярные выражения</h3>
```

```
<?php
```

```
$name = "Имя";
```

```
echo preg_match("/^[a-яА-Я ]*$/", $name);
```

```
echo "<hr>";
```

```
$email = "w@mail.ru";
```

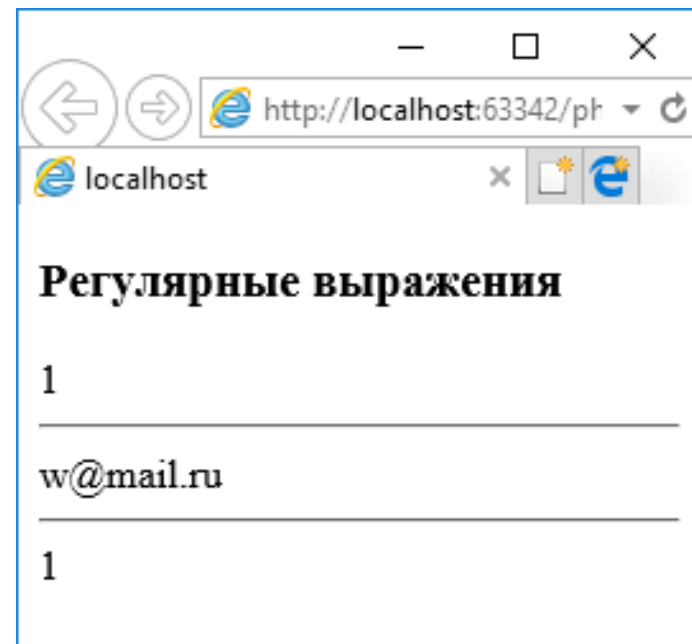
```
echo filter_var($email, FILTER_VALIDATE_EMAIL);
```

```
echo "<hr>";
```

```
$website = "http://asd";
```

```
echo preg_match("/\b(?:(:https?|ftp):\\/\|www\.)[-a-z0-9+&@#\\/%?~_!|:.,;]*[-a-z0-9+&@#\\/%?~_]|/i", $website);
```

```
?>
```



# Многомерные массивы

## demo23.php

<h3>Многомерные массивы</h3>

<?php

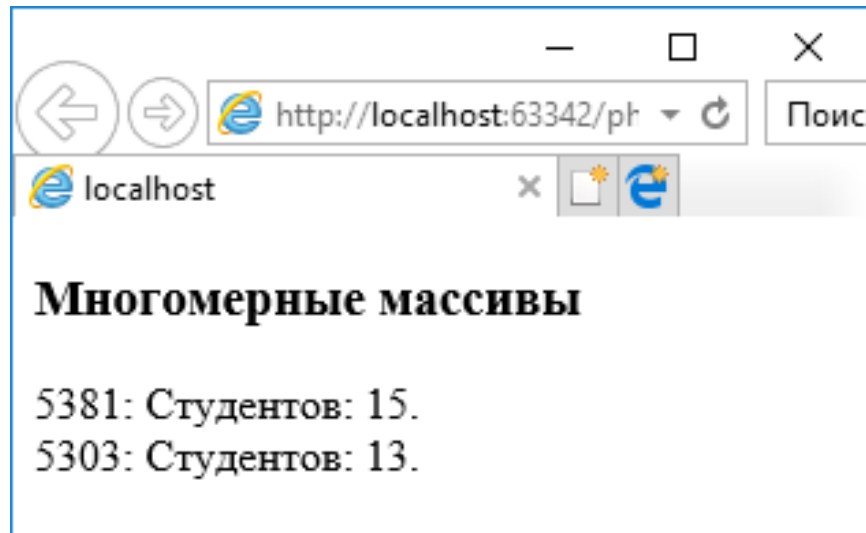
```
$groups = array (  
    array("5381",15),  
    array("5303",13),
```

```
);
```

```
echo $groups[0][0].": Студентов: ".$groups[0][1].".<br>;
```

```
echo $groups[1][0].": Студентов: ".$groups[1][1].".<br>;
```

```
?>
```



# Даты

## demo24.php

```
<h3>Даты</h3>
```

```
<?php
```

```
echo "Сегодня " . date("Y-m-d") . "<br>";
```

```
echo "Сегодня " . date("d.m.Y") . "<br>";
```

```
echo "Сегодня " . date("Y") . "<br>";
```

```
$d=mktime(11, 13, 12, 8, 12, 2017);
```

```
echo "Дата " . date("d.m.Y h:i:sa", $d) . "<br>";
```

```
$d=strtotime("10:30pm December 12 2017");
```

```
echo "Дата " . date("d.m.Y h:i:sa", $d) . "<br>";
```

```
$d=strtotime("tomorrow");
```

```
echo "Завтра " . date("d.m.Y h:i:sa", $d) . "<br>";
```

```
$d=strtotime("next Saturday");
```

```
echo "Следующая суббота " . date("d.m.Y h:i:sa", $d) . "<br>";
```

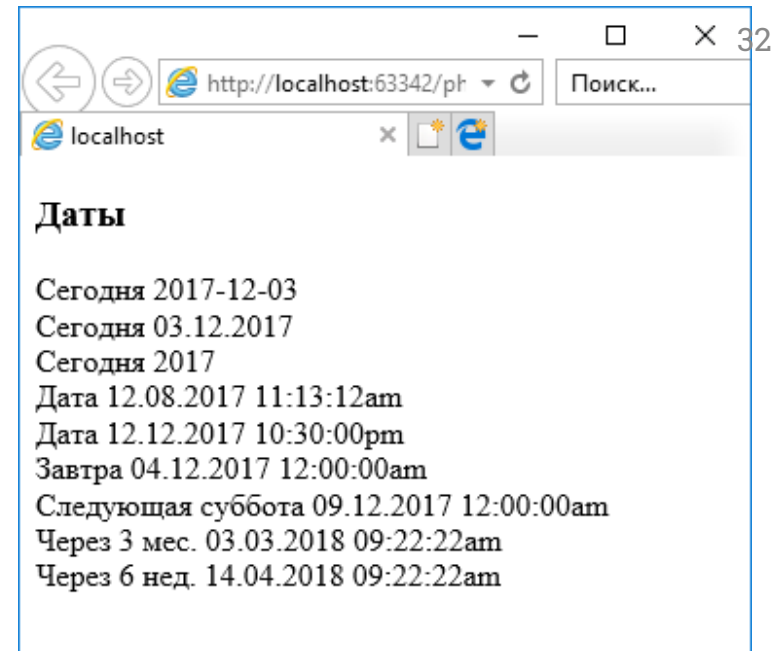
```
$d=strtotime("+3 Months");
```

```
echo "Через 3 мес. " . date("d.m.Y h:i:sa", $d) . "<br>";
```

```
$d = strtotime("+6 weeks", $d);
```

```
echo "Через 6 нед. " . date("d.m.Y h:i:sa", $d) . "<br>";
```

```
?>
```



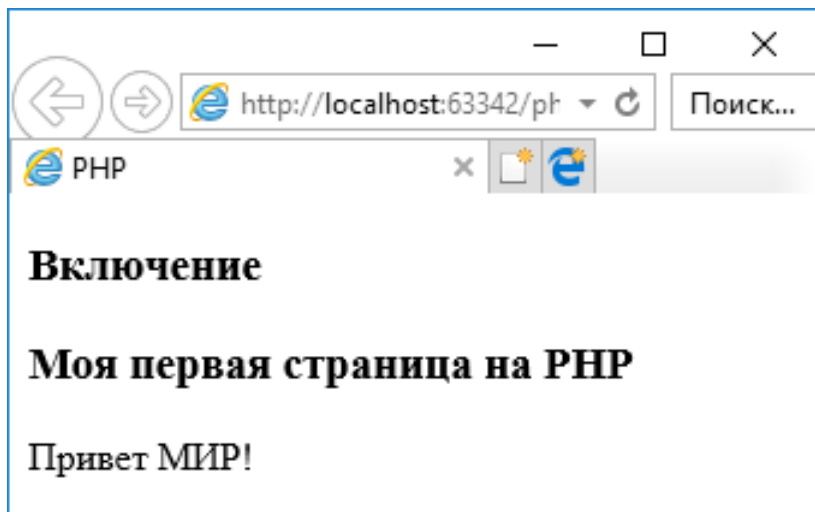


# Include / require страниц

demo25.php

<h3>Включение</h3>

```
<?php include "demo01.php"
?>
```



- Включение может осуществляться с использованием
  - **include**
  - **require**
- В случае, если файл не существует
  - **include** продолжит выполняться
  - **require** прекратит выполнение с ошибкой

# Простейшее чтение файла

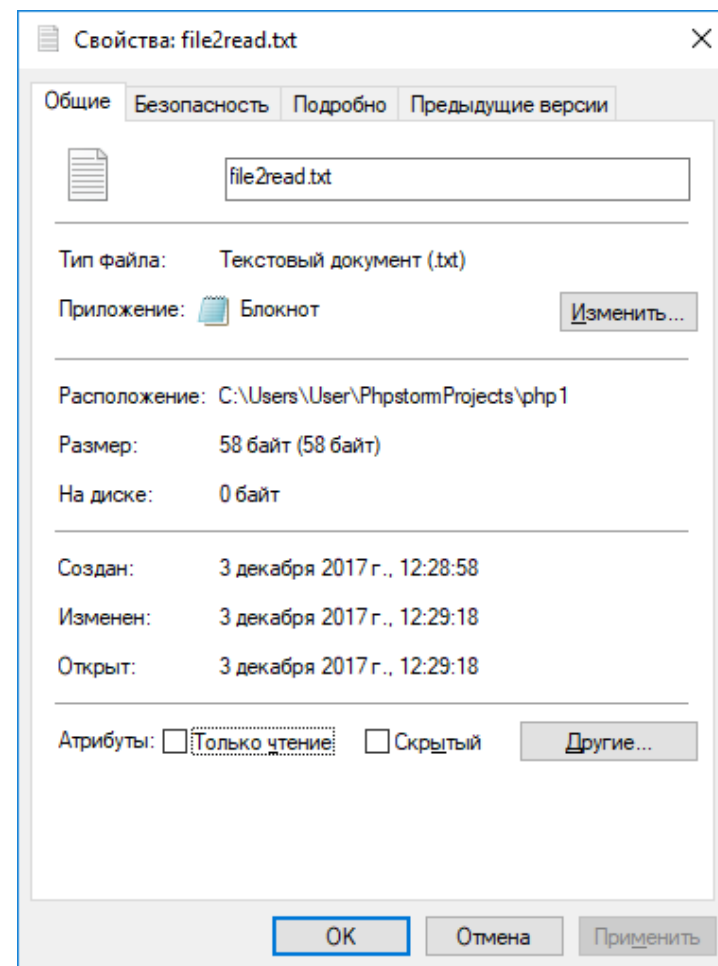
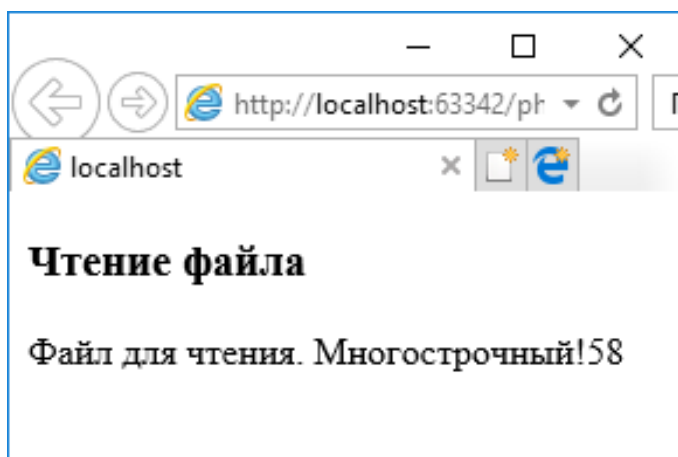
34

## demo26.php

```
<?php  
echo readfile("file2read.txt");  
?>
```

## file2read.txt

Файл для чтения.  
Многострочный!



# Режимы работы с файлами

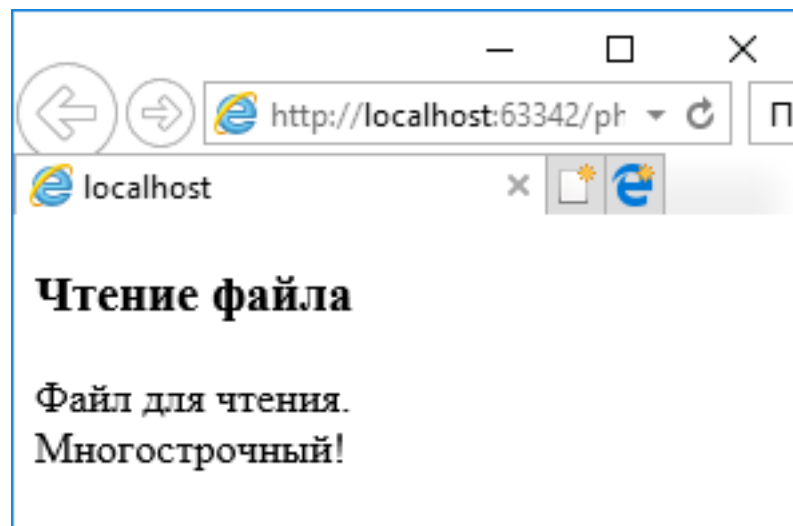
Режим	Описание
r	Открывает файл только для чтения
w	Открывает файл только для записи, создаёт и перезаписывает содержимое
a	Открывает файл только для записи с сохранением записанных данных
x	Создаёт новый файл для записи
r+	Создаёт файл для чтения и записи
w+	Создаёт файл для чтения и записи с перезаписыванием содержимого

# Чтение файла построчно

## demo27.php

<h3>Чтение файла</h3>

```
<?php
$myfile = fopen("file2read.txt", "r") or die("Не открыть файл!");
// Открывает одну строку до конца файла
while(!feof($myfile)) {
    echo fgets($myfile) . "<br>";
}
fclose($myfile);
?>
```



# Чтение файла посимвольно

37

## demo28.php

```
<h3>Чтение файла</h3>
```

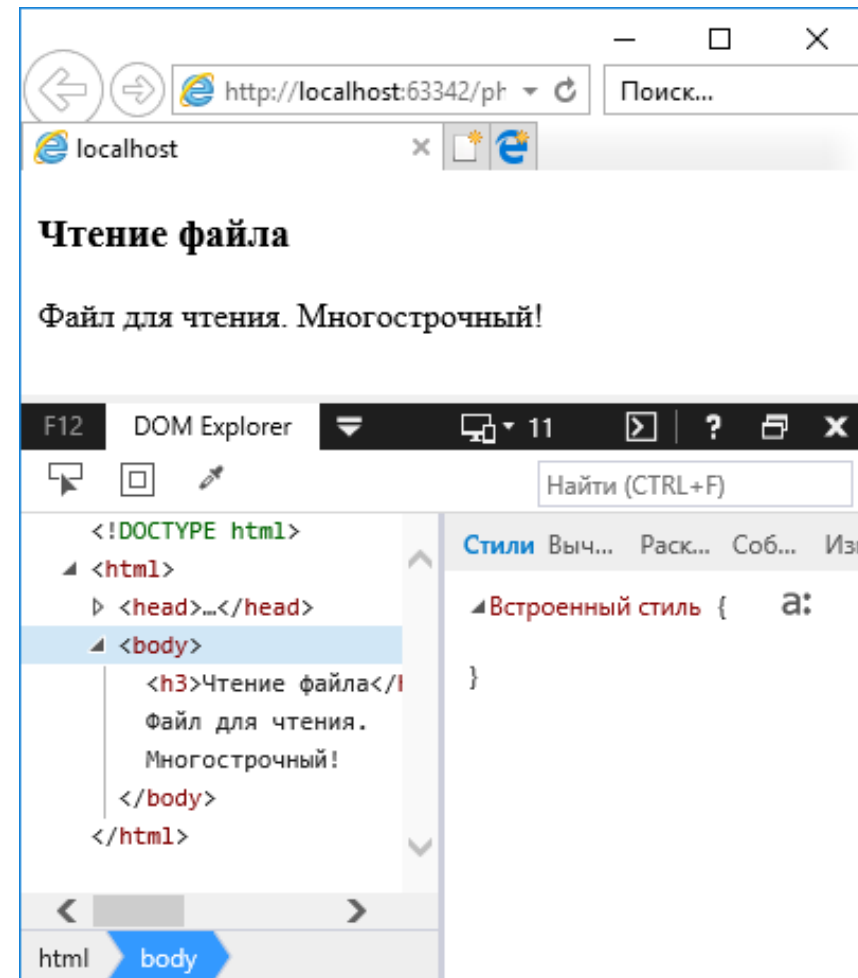
```
<?php
```

```
$myfile = fopen("file2read.txt", "r") or die("Не открыть файл!");
```

```
while(!feof($myfile)) {  
    echo fgetc($myfile);  
}
```

```
fclose($myfile);
```

```
?>
```



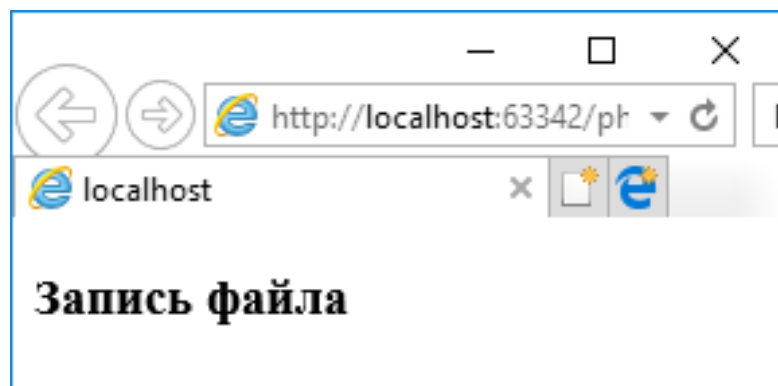
# Запись файла

## demo29.php

```
<?php
$myfile = fopen("newfile.txt", "w") or die("Не открыть файл!");
$txt = "Web-технологии\n";
fwrite($myfile, $txt);
$txt = "2017 год и далее...\n";
fwrite($myfile, $txt);
fclose($myfile);
?>
```

## newfile.txt

Web-технологии  
2017 год и далее...



# Загрузка файла (1)

39

## demo30.php

```
<?php
$target_dir = "uploads/";
$target_file = $target_dir . basename($_FILES["fileToUpload"]["name"]);
$uploadOk = 1;
$imageFileType = pathinfo($target_file,PATHINFO_EXTENSION);
// Проверить, что это действительно картинка
if(isset($_POST["submit"])) {
    $check = getimagesize($_FILES["fileToUpload"]["tmp_name"]);
    if($check !== false) {
        echo "File is an image - " . $check["mime"] . ".";
        $uploadOk = 1;
    } else {
        echo "File is not an image.";
        $uploadOk = 0;
    }
}
// Проверить, что файл существует
if (file_exists($target_file)) {
    echo "Sorry, file already exists.";
    $uploadOk = 0;
}
// Проверить размер файла
if ($_FILES["fileToUpload"]["size"] > 500000) {
    echo "Sorry, your file is too large.";
    $uploadOk = 0;
}

// Разрешить определённый формат
if($imageFileType != "jpg" && $imageFileType != "png"
    && $imageFileType != "jpeg"
    && $imageFileType != "gif" ) {
    echo "Sorry, only JPG, JPEG, PNG & GIF files are
allowed.";
    $uploadOk = 0;
}
// Проверить, что можно загружать
if ($uploadOk == 0) {
    echo "Sorry, your file was not uploaded.";
    // Если всё нормально, загружаем
} else {
    if
(move_uploaded_file($_FILES["fileToUpload"]["tmp_name"],
    $target_file)) {
        echo "The file " . basename(
$_FILES["fileToUpload"]["name"]) .
        " has been uploaded.";
    } else {
        echo "Sorry, there was an error uploading your
file.";
    }
}
}??>
```

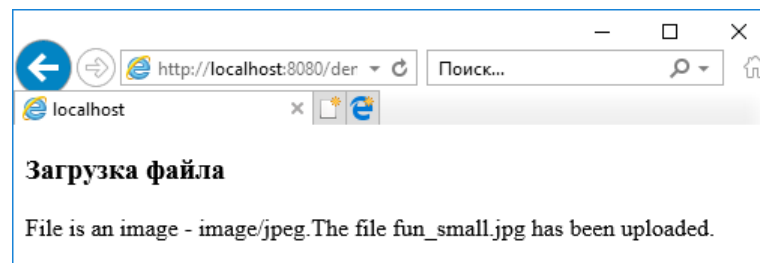
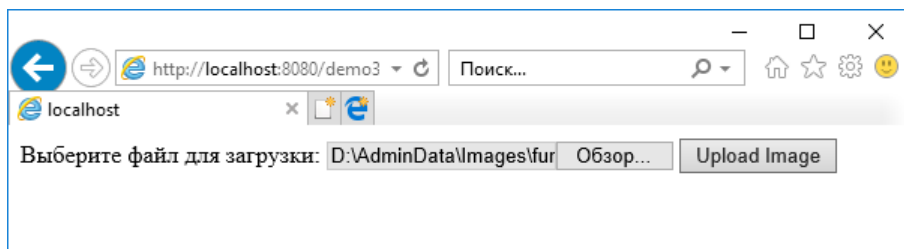
# Загрузка файла (2)

## demo30.html

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
</head>
<body>

<form action="demo30.php" method="post" enctype="multipart/form-data">
  Выберите файл для загрузки:
  <input type="file" name="fileToUpload" id="fileToUpload">
  <input type="submit" value="Upload Image" name="submit">
</form>

</body>
</html>
```





# Работа с cookie

41

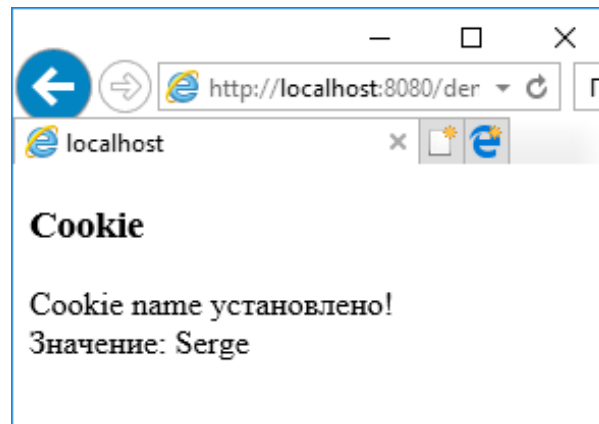
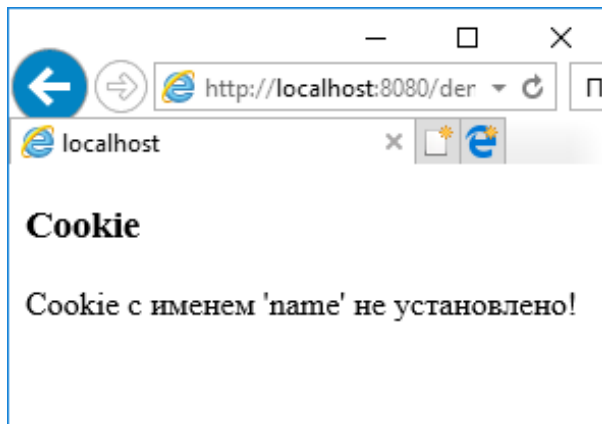
## demo31.php

```
<?php
$cookie_name = "name";
$cookie_value = "Serge";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>
```

```
<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie с именем '" . $cookie_name . "' не установлено!";
} else {
    echo "Cookie $cookie_name установлено!<br>";
    echo "Значение: " . $_COOKIE[$cookie_name];
}
?>
```

## Синтаксис:

setcookie(name, value, expire, path, domain, secure, httponly);

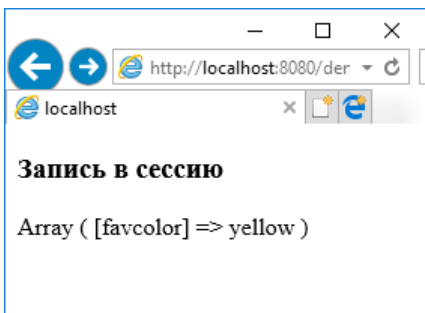


## demo32.php

```
<?php
session_start();
?>
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>

<h3>Запись в сессию</h3>
<?php
$_SESSION["favcolor"] = "yellow";
print_r($_SESSION);
?>

</body>
</html>
```

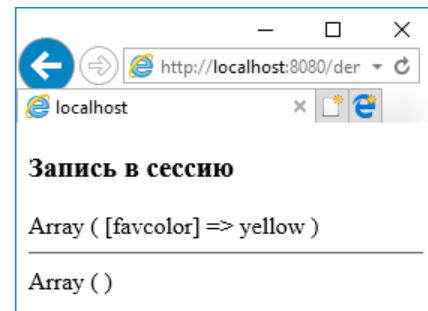


## demo32b.php

```
<?php
session_start();
?>
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
</head>
<body>

<h3>Запись в сессию</h3>
<?php
print_r($_SESSION);
// Удалить все переменные
session_unset();
// УНИЧТОЖИТЬ сессию
session_destroy();
echo "<hr>";
print_r($_SESSION);
?>

</body>
</html>
```



# Использование фильтров для валидации

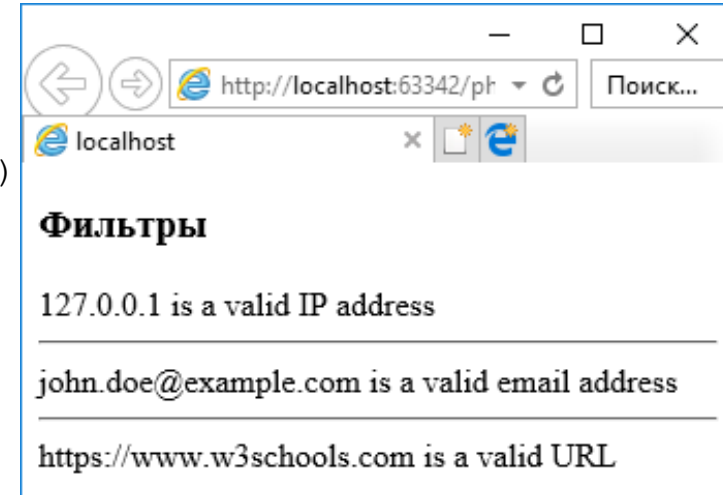
43

## demo33.php

```
<?php
$ip = "127.0.0.1";
if (!filter_var($ip, FILTER_VALIDATE_IP) === false)
    echo("$ip is a valid IP address");
else
    echo("$ip is not a valid IP address");
echo "<hr>";

$email = "john.doe@example.com";
// Удаление всех запрещённых символов из email
$email = filter_var($email, FILTER_SANITIZE_EMAIL);
// Валидация e-mail
if (!filter_var($email, FILTER_VALIDATE_EMAIL) === false)
    echo("$email is a valid email address");
else
    echo("$email is not a valid email address");
echo "<hr>";

$url = "https://www.w3schools.com";
// Удаление всех запрещённых символов из url
$url = filter_var($url, FILTER_SANITIZE_URL);
// Валидация url
if (!filter_var($url, FILTER_VALIDATE_URL) === false)
    echo("$url is a valid URL");
else
    echo("$url is not a valid URL");
?>
```



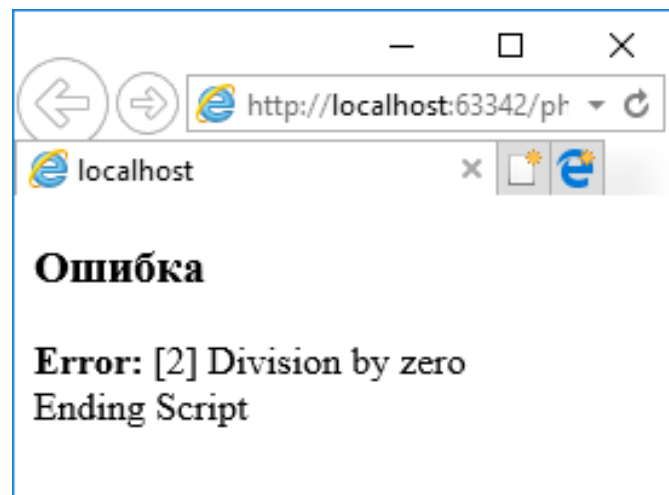
# Обработка ошибок

44

## demo34.php

```
<?php
function customError($errno, $errstr) {
    echo "<b>Error:</b> [$errno] $errstr<br>";
    echo "Ending Script";
    die();
}
set_error_handler("customError");

function throwException() {
    try{
        $x = 12 / 0;
    } catch (Exception $e) {
        echo "Поймал ошибку<br>" . $e->getMessage();
    }
}
throwException();
?>
```



### Синтаксис:

`error_function(error_level, error_message, error_file, error_line, error_context)`

# Уровни ошибок

Значение	Константа	Описание
2	E_WARNING	Не фатальные ошибки, выполнение скрипта продолжится
8	E_NOTICE	Предупреждения
256	E_USER_ERROR	Фатальные ошибки, созданные пользователем
512	E_USER_WARNING	Не фатальные предупреждения, созданные пользователем
1024	E_USER_NOTICE	Созданные пользователем нотификации
4096	E_RECOVERABLE_ERROR	Фатальные ошибки, которые могут быть «пойманы»
8191	E_ALL	Все ошибки и предупреждения

# Исключительная ситуация (1)

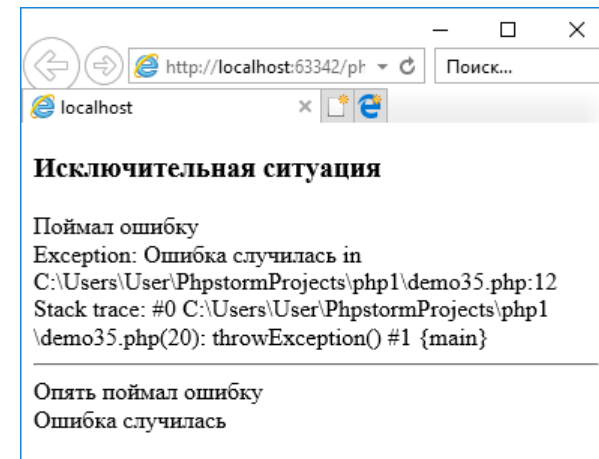
46

## demo35.php

<h3>Исключительная ситуация</h3>

<?php

```
function throwException() {  
    try{  
        throw new Exception("Ошибка случилась");  
    } catch(Exception $e) {  
        echo "Поймал ошибку<br>" . $e;  
        echo "<hr>";  
        throw $e;  
    }  
}  
  
try{  
    throwException();  
} catch(Exception $e) {  
    echo "Опять поймал ошибку<br>" . $e->getMessage();  
}  
?>
```

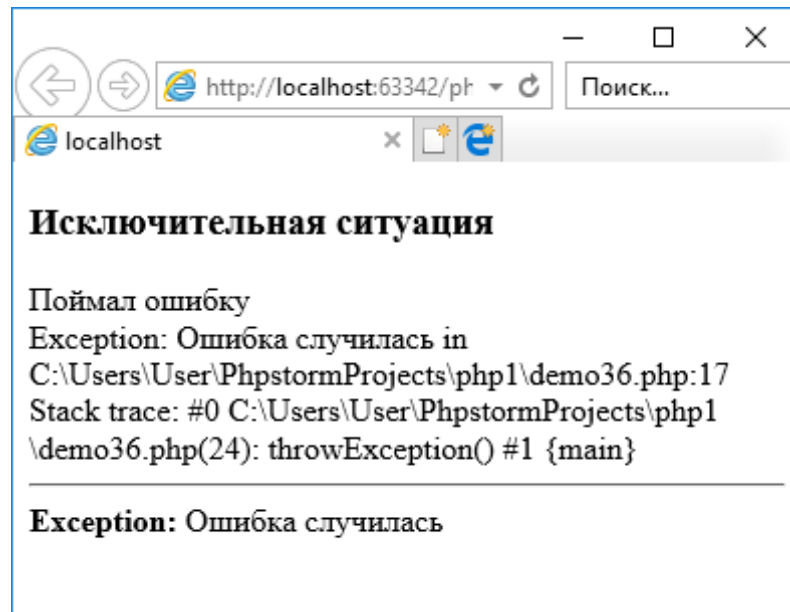


# Исключительная ситуация (2) – универсальный обработчик

## demo36.php

```
<?php
function myException($exception) {
    echo "<b>Exception:</b> " . $exception->getMessage();
}
set_exception_handler('myException');

function throwException() {
    try{
        throw new Exception("Ошибка случилась");
    } catch (Exception $e) {
        echo "Поймал ошибку<br>" . $e;
        echo "<hr>";
        throw $e;
    }
}
throwException();
?>
```



# Обработка XML

48

## demo37.php

```
<?php
```

```
$myfile = fopen("test.xml", "r") or die("Не открыть файл!");
```

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

```
while(!feof($myfile)) {  
    $str .= fgets($myfile);  
}  
fclose($myfile);
```

```
$xml = simplexml_load_string($str);
```

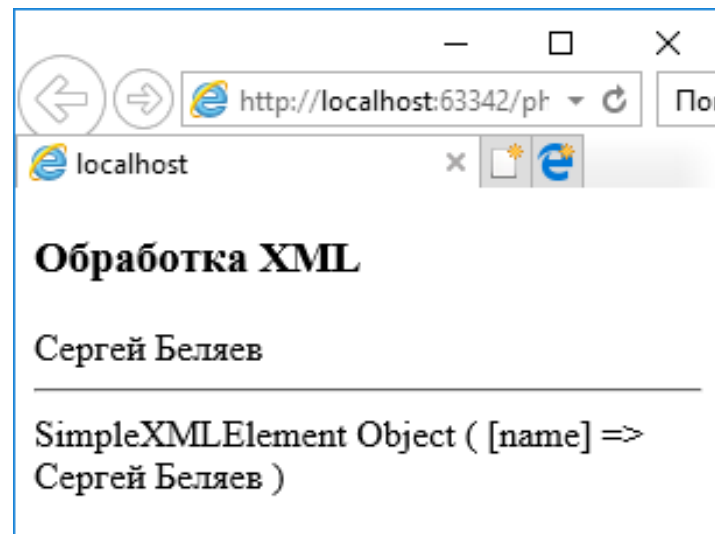
```
if ($xml === false) {  
    echo "Failed loading XML: ";  
    foreach(libxml_get_errors() as $error) {  
        echo "<br>", $error->message;  
    }  
}
```

```
else {  
    echo $xml->name;  
    echo "<hr>";  
    print_r($xml);  
}
```

```
?>
```

## test.xml

```
<?xml version='1.0' encoding='UTF-8' ?>  
<test>  
    <name>Сергей Беляев</name>  
</test>
```



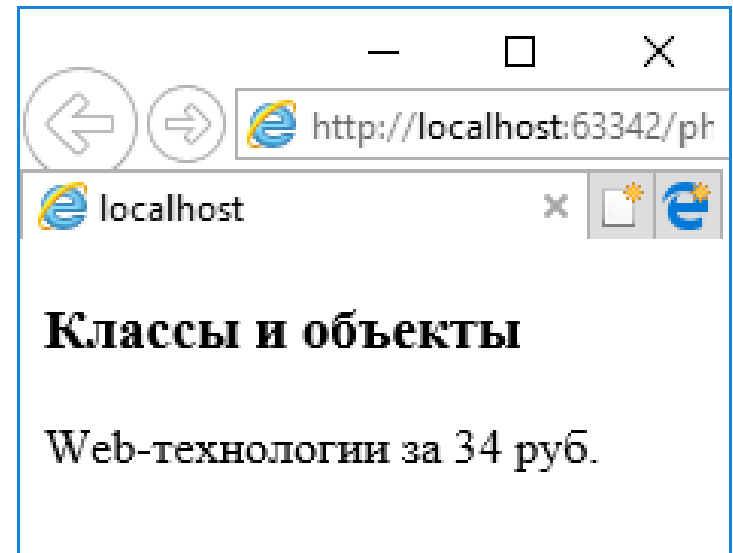


## demo38.php

```
<?php
class BooksFixedPrice {
    var $price = 34;
    var $title;

    function setTitle($par){
        $this->title = $par;
    }
    function getTitle(){
        echo $this->title;
    }
}

$book = new BooksFixedPrice();
$book->setTitle("Web-технологии");
echo $book->getTitle() . " за " . $book->price . " руб.";
?>
```



- Проблемы со строками
  - Нет поддержки юникода. Надёжно работает только ASCII
    - Это означает, что использование встроенных строковых функций с UTF-8-текстом создаёт риск их порчи
  - Нет концепции типа сравнения регистра вне ASCII
  - Ключи нельзя кватировать при интерполяции переменных
    - `"${foo['key']}"` — ошибка синтаксиса
  - `"${foo[0]}"` работает
    - `"${foo[0][0]}"` — синтаксическая ошибка

- **==**
  - Не транзитивен
    - `"foo" == TRUE`, и `"foo" == 0`, но `TRUE != 0`.
  - `==` конвертирует в число, если возможно, далее конвертирует в `float`, если возможно
    - Большие шестнадцатеричные строки (например, хеши паролей) могут неожиданно быть равными, когда они не равны
  - `"6" == " 6"`, `"4.2" == "4.20"` и `"133" == "0133"`
    - Но `133 != 0133`, потому что `0133` восьмеричное.
  - `===` сравнивает значения и тип, но не для объектов, где `===` истинно если оба операнда один и тот же объект
  - Для объектов `==` сравнивает оба значения (для каждого атрибута) и типы

# Критика PHP (3)

52

- `[]` оператор индекса может также быть записан как `{ }`
- `[]` может быть применён к любой переменной, не только к строкам и массивам
  - Он возвращает **null** и не выдаёт предупреждение
- В отличие от (почти) всех остальных языков с похожим оператором, `?:` левоассоциативен
- Имена переменных чувствительны к регистру, а имена функций и классов нет
- **array()** и пару дюжин других конструкций — не функции
  - Сама по себе конструкция **array** не означает ничего, `$func = "array"; $func();` не работает
- Оператор `@` заглушает ошибки
- У ошибок PHP **нет стэктрейсов**
  - Вы должны установить обработчик, чтобы их генерировать
- **=>** — не оператор. Это специальная конструкция, существующая только внутри конструкций **array(...)** и **foreach**

- Можно

- попытаться получить **доступ к несуществующему свойству** объекта, типа `$foo->x` (warning)
- попытаться использовать **неопределённую константу** (notice)
- попытаться получить **доступ к свойству** чего-нибудь, не являющегося объектом (notice)
- попытаться использовать **имя переменной, которая не существует** (notice)
- `2 < "foo"` (без сообщений)
- `foreach (2 as $foo);` (warning)

- Нельзя

- попытаться получить **доступ к несуществующей константе** класса, типа `$foo::x` (fatal error)
- попытаться **вызвать неопределённую функцию** (fatal error)
- **оставлять точку с запятой** в конце блока или файла (parse error)
- использовать `list` и другие квазиконструкции как названия методов (parse error)

# Система управления содержимым

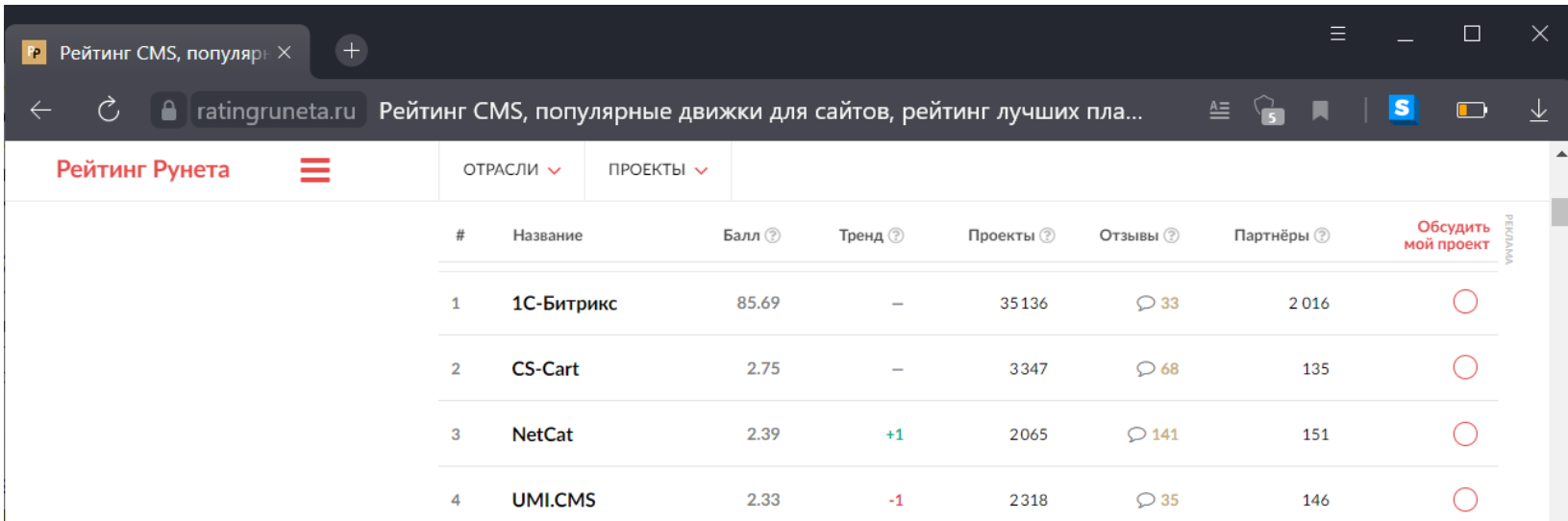
54

- Обеспечение и организация совместного процесса создания, редактирования и управления содержимым
- Основные функции
  - Предоставление инструментов для
    - создания содержимого
    - организации совместной работы
  - Управление содержимым
    - хранение, контроль версий, соблюдение режима доступа и т. п.
  - Публикация содержимого
  - Представление информации в виде, удобном для навигации и поиска
- Типичные модули
  - меню
  - блог
  - новости
  - опросы
  - поиск по сайту
  - статистика посещений
  - гостевая книга

# CMS – Content management system

55

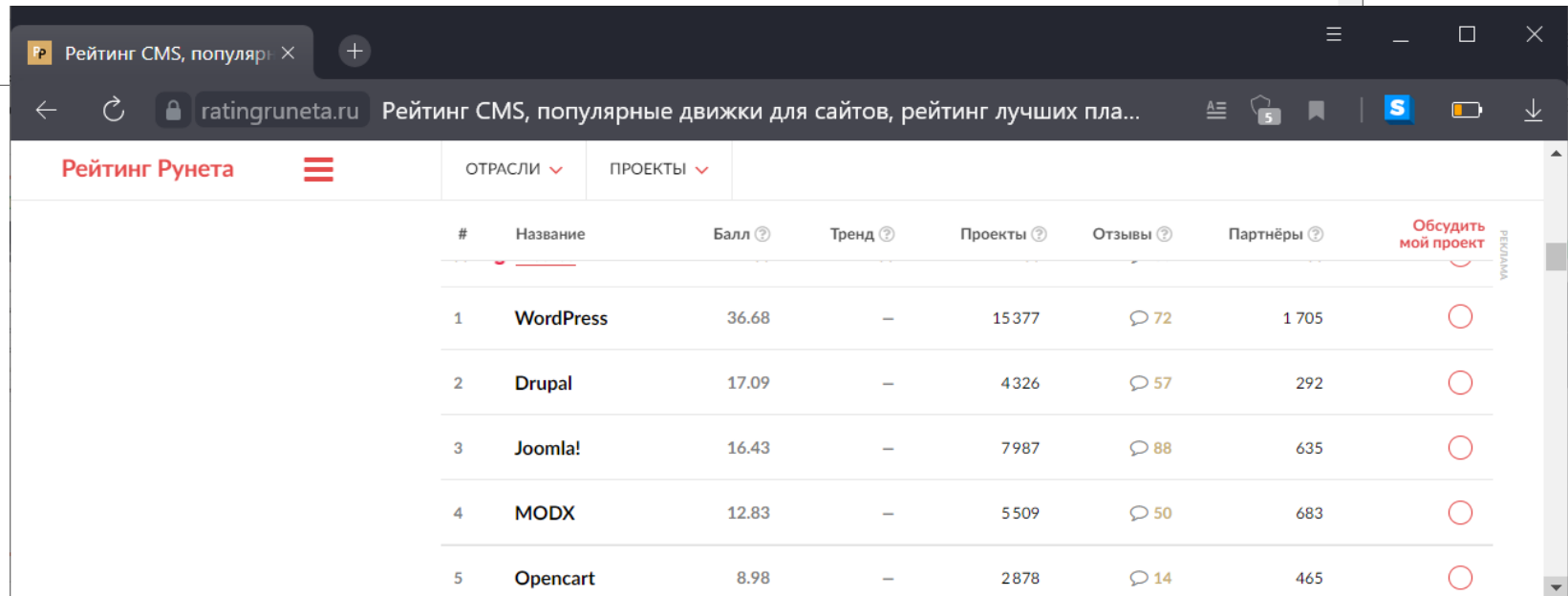
Коммерческие



Рейтинг Рунета

#	Название	Балл ?	Тренд ?	Проекты ?	Отзывы ?	Партнёры ?	Обсудить мой проект
1	1С-Битрикс	85.69	—	35136	33	2 016	
2	CS-Cart	2.75	—	3347	68	135	
3	NetCat	2.39	+1	2065	141	151	
4	UMI.CMS	2.33	-1	2318	35	146	

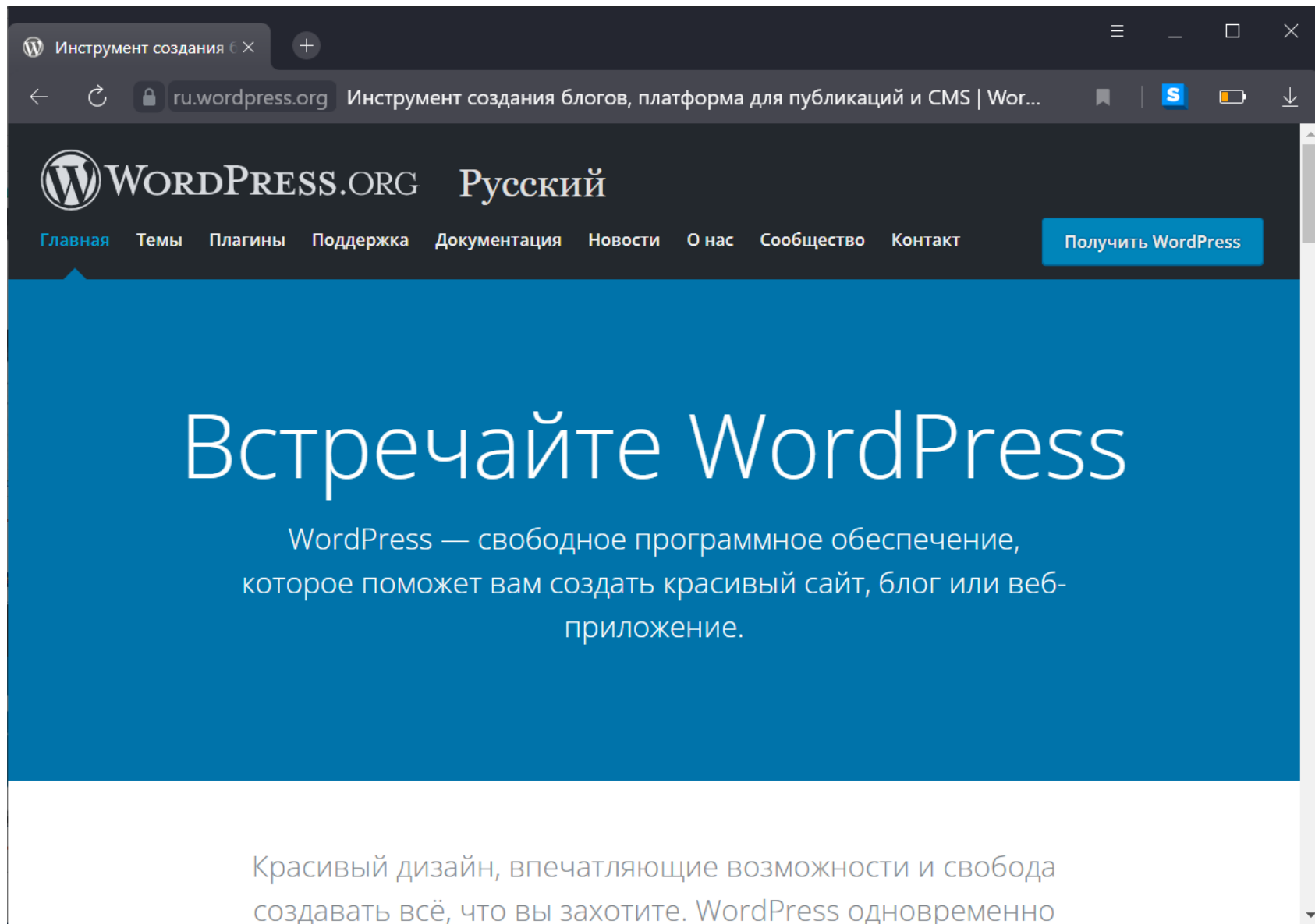
Бесплатные



Рейтинг Рунета

#	Название	Балл ?	Тренд ?	Проекты ?	Отзывы ?	Партнёры ?	Обсудить мой проект
1	WordPress	36.68	—	15377	72	1 705	
2	Drupal	17.09	—	4326	57	292	
3	Joomla!	16.43	—	7987	88	635	
4	MODX	12.83	—	5509	50	683	
5	Opencart	8.98	—	2878	14	465	

<https://ratingruneta.ru/cms/>



<https://ru.wordpress.org/>



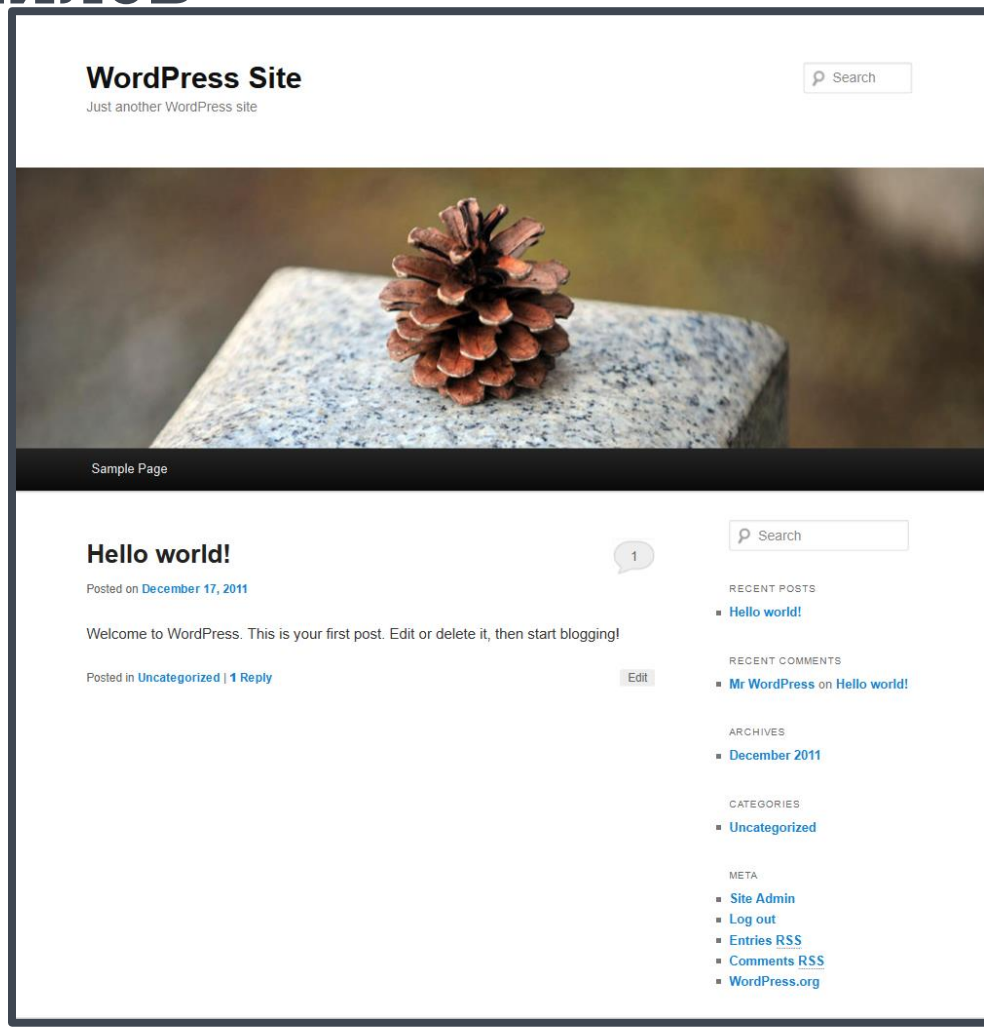
# WordPress. Управление и администрирование

- Локальная установка
- Переносимая основа
- Совместимость с UTC
- Поддержка gzip
- Управление пользователями
- Профили пользователей
- Простая установка и обновление
- Динамическая генерация страниц
- Интернационализация и локализация

# WordPress. Оформление дизайна

58

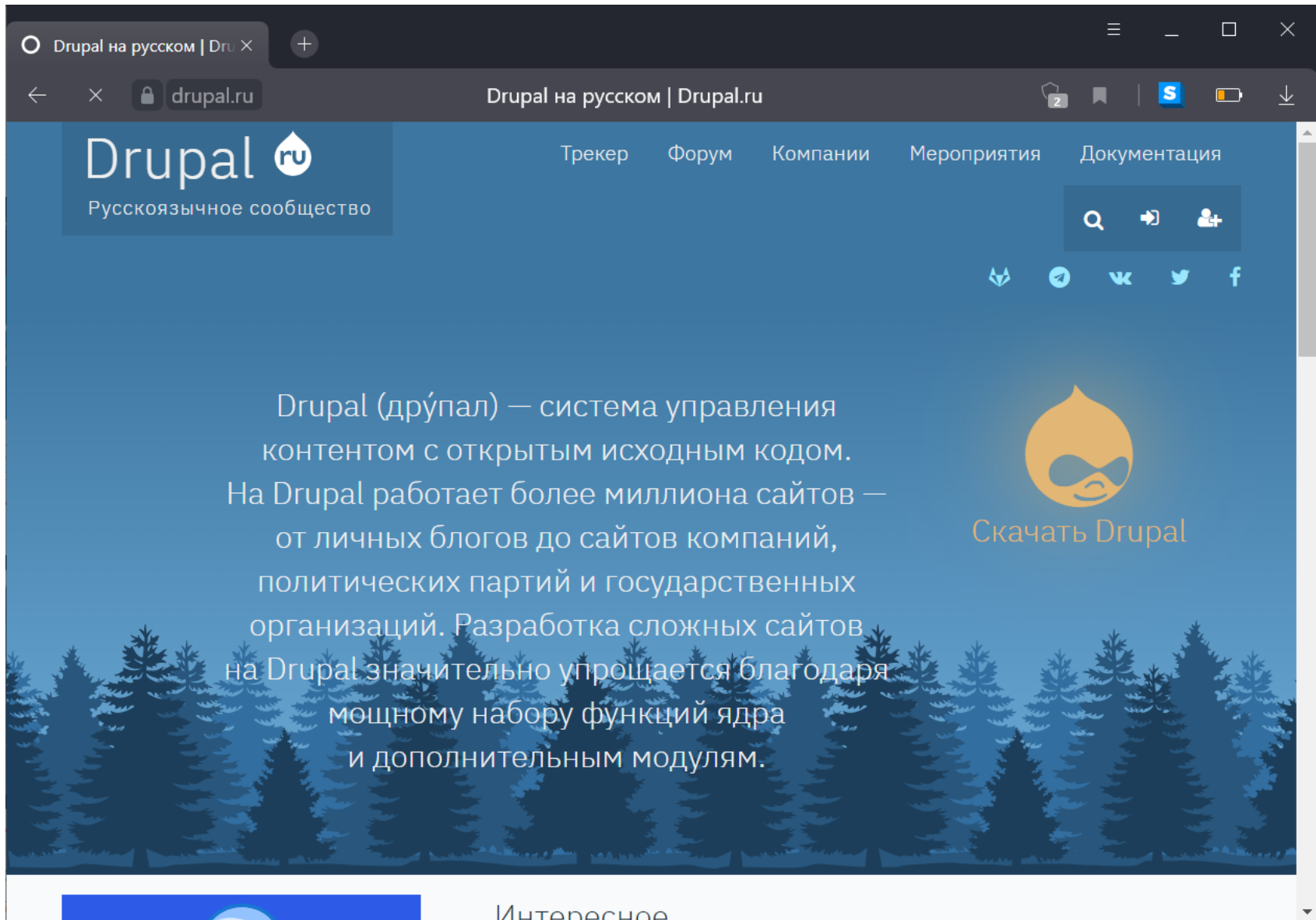
- Шаблоны управляют дизайном
- Шаблон и Редактор файлов
- Теги шаблонов
- Темы
- Плагины



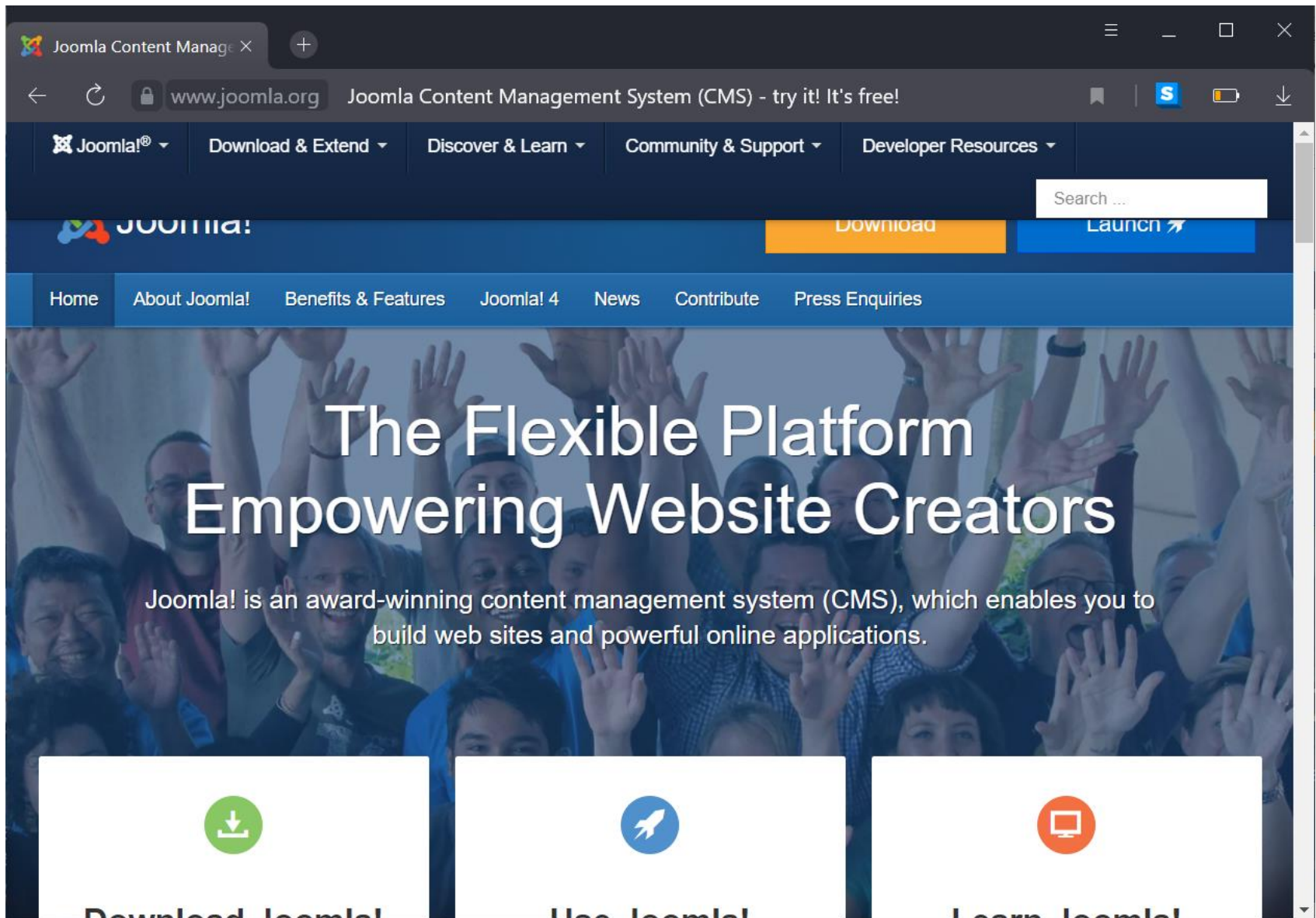
# WordPress. Создание содержания

59

- Защита паролем
- Индивидуальные адреса страниц
- Написать в будущем
- Многостраничные статьи
- Загрузка файлов
- Категории
- Смайлы
- Черновики
- Предпросмотр статей
- Публикация через специальные программы
- Публикация через email
- Bookmarklets
- Боковая панель
- Форматирование



Интересное  
<https://drupal.ru/>



<https://www.joomla.org/>

# Вопросы для самопроверки

62

- Как настроить PHP под Windows и Linux?
- Где выполняется PHP (сервер, клиент)?
- На какой "движок" шаблонов похож код на PHP? Чем отличается?
- Можно ли на PHP-страницах использовать JavaScript? Что и где при этом будет исполняться?
- Чем похожи и чем отличаются ключевые слова и правила написания названий переменных PHP и JavaScript?
- Чем похожи и чем отличаются шаблоны строк в JavaScript и в PHP?
- Как объявить переменную в PHP? Каких типов бывают переменные в PHP?
- Существуют ли в PHP константы?
- Чем похожи и чем отличаются операторы в JavaScript и в PHP?
- Какие суперглобальные переменные PHP вы знаете?
- Чем отличаются include и require?
- В чём отличие в работе с Cookie в JavaScript и PHP?
- Как обрабатывать ошибки в PHP?
- Чем опасна функция `phpinfo()`?
- Что такое CMS? Для чего они нужны?
- В каких случаях следует использовать CMS?