

Практическая работа №6 «PHP: работа с файлами»

Цель работы: познакомиться с различными приёмами работы с файлами в языке PHP.

ЗАДАНИЯ

А. Подготовка к работе:

1. Запустите менеджер виртуальных машин VMware Player на **реальном** компьютере.
2. В программе VMware Player запустите образ **виртуального** компьютера **WWW** с предустановленным ПО: Apache (HTTP-сервер), PHP (интерпретатор) и MySQL (СУБД).
3. В **адресной строке браузера** с реального компьютера откройте поочередно ссылки <http://www/>, <http://www/info.php> и <http://www/sqltest.php>. Убедитесь в работоспособности перечисленных выше программных компонентов виртуального веб-сервера.
4. В **проводнике на реальном** компьютере введите адрес [\\www](http://www), укажите по запросу имя пользователя – Администратор; пароль – 1234. Отобразится список общих папок веб-сервера, в т.ч. папка www – **корневая папка сайта** (локальный путь – C:\wamp\www).

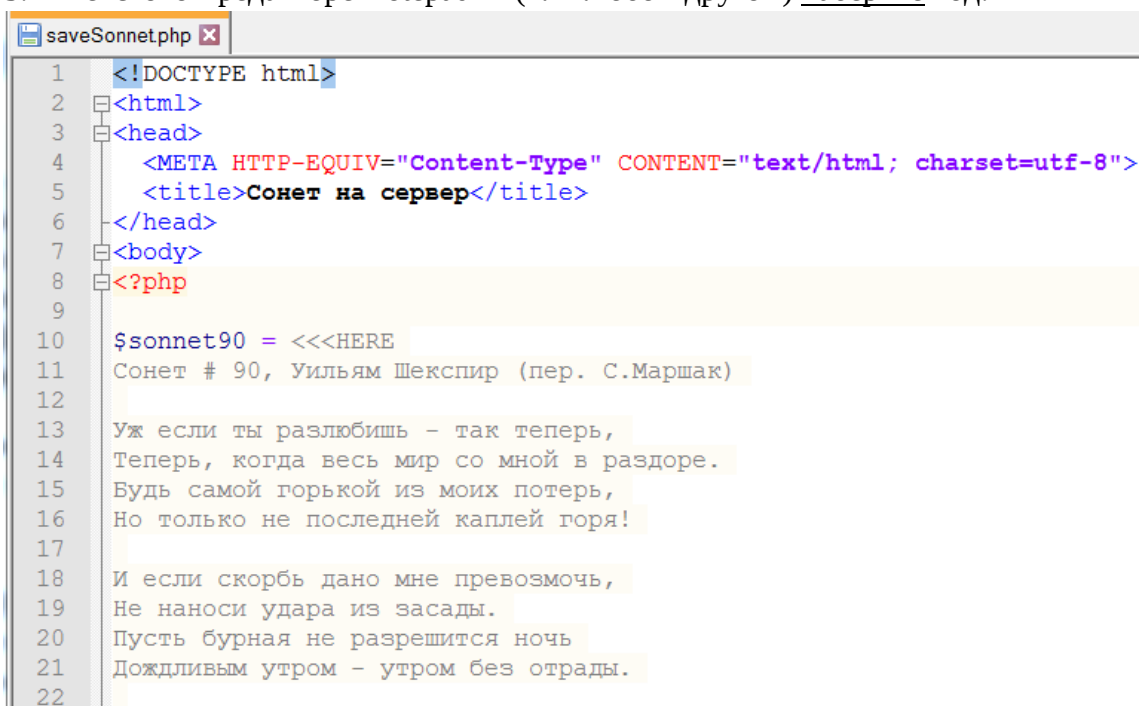
Если проводник сообщает, что нет доступа к ресурсу, то на реальном компьютере (для ОС Windows Vista / 7 / 8.x / 10) откройте «Панель управления» → «Сеть и Интернет» → «Центр управления сетями и общим доступом» → «Изменить дополнительные параметры общего доступа». В нужном профиле выберите «Включить сетевое обнаружение» и «Включить общий доступ, чтобы сетевые пользователи могли читать и записывать файлы в общих папках». Сохраните изменения и повторите попытку.

Возможной причиной блокирования также может быть **пустой пароль** у текущего пользователя реального компьютера.

Примечание: если запрещено изменение настроек на реальном компьютере, редактирование файлов сайта в дальнейшем осуществляйте прямо на виртуальном компьютере.

Б. Создание файла и запись в файл на сервере

5. В текстовом редакторе Notepad++ (или любом другом) наберите код:



```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
5      <title>Сонет на сервер</title>
6  </head>
7  <body>
8      <?php
9
10     $sonnet90 = <<<HERE
11     Сонет # 90, Уильям Шекспир (пер. С.Маршак)
12
13     Уж если ты разлюбишь - так теперь,
14     Теперь, когда весь мир со мной в раздоре.
15     Будь самой горькой из моих потерь,
16     Но только не последней каплей горя!
17
18     И если скорбь дано мне превозмочь,
19     Не наноси удара из засады.
20     Пусть бурная не разрешится ночь
21     Дожливым утром - утром без отрады.
22

```

```

23 Оставь меня, но не в последний миг,
24 Когда от мелких бед я ослабею.
25 Оставь сейчас, чтоб сразу я постиг,
26 Что это горе всех невзгод больнее,
27
28 Что нет невзгод, а есть одна беда –
29 Твоей любви лишиться навсегда.
30 HERE;
31
32 $fp = fopen("sonnet90.txt", "w");
33 fputs($fp, $sonnet90);
34 fclose($fp);
35 ?>
36 </body>
37 </html>

```

Сохраните код в файле saveSonnet.php.

Программа создаёт файл на сервере и записывает в него текст.

Функция fopen() открывает (создаёт в случае отсутствия) файл на сервере.

Примечание: файл на клиентском компьютере не может быть создан – у серверного скрипта нет доступа к клиентской файловой системе.

Первым параметром функции является имя файла, включающее в себя абсолютный или относительный путь в папке на сервере, или унифицированный указатель. Второй параметр определяет способ открытия файла (режим доступа) – чтение и/или запись, очищение файла или добавление в него.

Сама функция fopen() возвращает: а) указатель на открываемый файл (дескриптор файла), который в данном примере помещается в переменную \$fp; или б) значение FALSE в случае ошибки.

Ознакомьтесь, используя справочник по PHP, с режимами доступа к файлу.

Функция fputs() записывает содержимое переменной \$sonnet90 в файл по указателю из переменной \$fp.

Функция fclose() закрывает файл, т.е. сообщает операционной системе, что скрипт завершил работу с этим файлом.

Примечание: очень важно не забывать о явном закрытии файла. В наше время сложных технологий буферизации и оптимизации дисковых операций это позволит избежать файловых ошибок и, как следствие, потерь данных.

Запустите скрипт. Найдите созданный на сервере файл, проверьте его содержимое.

Как изменится содержимое созданного файла, если запустить скрипт еще раз? Как открыть файл для записи без потери ранее записанной в нем информации?

6. Исправьте код скрипта так, чтобы контролировалась возможность возникновения ошибки и чтобы дальнейшие действия с файлом производились только при удачной попытке открытия.

В. Считывание из файла

7. Наберите код скрипта для загрузки и отображения текста из ранее созданного файла:

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
5   <title>Сонет 90</title>

```

```

6 <style type = "text/css">
7   body {
8     background-color:darkred;
9     color: white;
10    font-family: 'Brush Script MT', script;
11    font-size: 1.5em;
12    text-align: center; }
13 </style>
14 </head>
15
16 <body>
17 <div>
18 <?php
19   $fp = fopen("sonnet90.txt", "r");
20
21   //первая строка - заголовок
22   $line = fgets($fp);
23   print "<h1>$line</h1><hr>";
24
25   //сам сонет
26   while (!feof($fp)){
27     $line = fgets($fp);
28     print "$line <br>";
29   } // end while
30
31   fclose($fp);
32 >?
33 </div>
34 </body>
35 </html>

```

Проверьте работу скрипта в браузере.

Здесь с помощью функции `fopen()` созданный ранее текстовый файл открывается в режиме только для чтения. Функция `fgets()` позволяет считать из файла строку.

Считывание основного содержимого файла осуществляется в цикле, пока не будет достигнут конец файла – функция `feof()`. Последней командой файл закрывается.

Примечание: для открытия файла в PHP существуют и другие команды. Так, уже использовавшаяся во второй работе функция `readfile()` позволяет считывать содержимое файла для вывода. Функция `file()` загружает содержимое файла в массив для его последующей обработки в скрипте.

Отредактируйте скрипт, учтя возможность возникновения ошибки.

Г. Работа с каталогами

8. Наберите код скрипта для формирования файла со списком имен файлов в заданном каталоге на сервере:

```

imageIndex.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
5   <title>Создание галереи</title>
6 </head>
7 <body>
8
9 <?php
10 $dirName = "pics"; // путь к каталогу с изображениями
11 $dp = opendir($dirName);
12 chdir($dirName);

```

```

13
14 //все файлы каталога - в массив
15 $currentFile = "";
16 while ($currentFile !== false){
17     $currentFile = readDir($dp);
18     $theFiles[] = $currentFile;
19 } // end while
20
21 //извлечь все gif и jpg
22 $imageFiles = preg_grep("/jpg$|gif$/", $theFiles);
23
24 $output = <<<HERE
25 <!DOCTYPE html>
26 <html>
27 <head>
28     <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
29     <title>Галерея</title>
30 </head>
31 <body>
32
33 <div>
34     HERE;
35
36 foreach ($imageFiles as $currentFile){
37     $output .= <<<HERE
38     <a href = "$currentFile">
39         <img src = "$currentFile"
40             height = "50"
41             width = "50"
42             alt = "$currentFile"></a>
43     HERE;
44 } // end foreach
45
46 $output .= <<<HERE
47 </div>
48 </body>
49 </html>
50 HERE;
51
52 //сохранение списка в файл
53 $fp = fopen("imageIndex.html", "w"); // имя файла - imageIndex.html
54 fputs ($fp, $output);
55 fclose($fp);
56
57 print <<<HERE
58 <p>Галерея
59 <a href = "$dirName/imageIndex.html">здесь</a>
60 </p>
61
62 HERE;
63 ?>
64
65 </body>
66 </html>

```

Сохраните код в файле imageIndex.php.

Создайте в папке сайта вложенную папку pics, скопируйте (или перенесите) в неё уже использовавшиеся в предыдущих работах файлы silly.gif, die1.jpg, ..., die6.jpg.

Проверьте работу скрипта. В папке pics на веб-сервере найдите созданный скриптом HTML-документ, просмотрите его содержимое в программе Блокнот.

Здесь функцией `opendir()` открывается каталог (возвращается его дескриптор), откуда с помощью функции `readdir()` в цикле считываются в массив все файлы этого каталога.

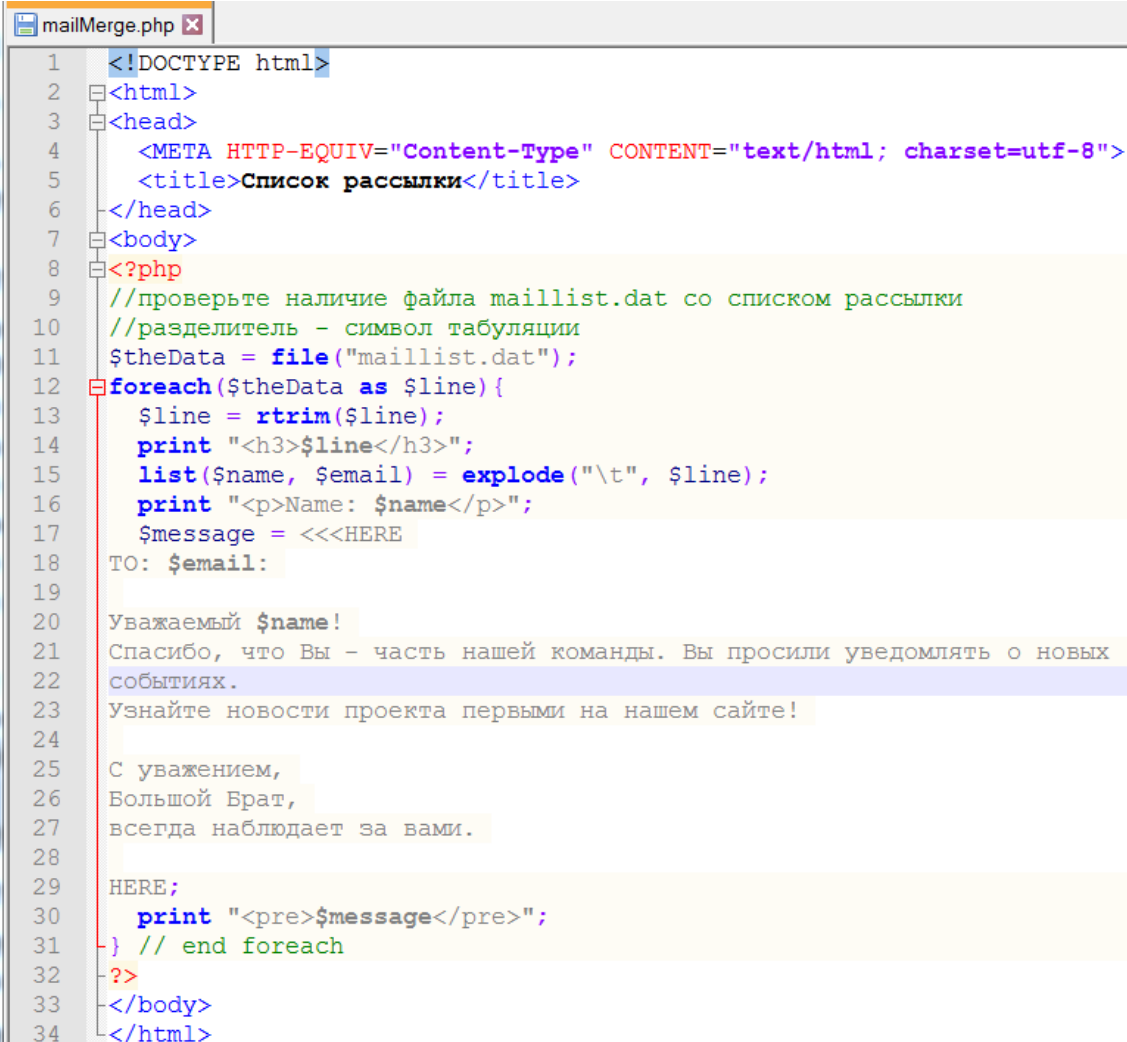
Функция `preg_grep()` позволяет фильтровать массив в соответствии с заданным шаблоном (Perl-совместимым *регулярным выражением* [PCRE](#)). В частности, в данном примере регулярное выражение начинается и заканчивается символом `/`, символ `|` означает операцию «или», знак `$` – конец символьной строки.

В переменной `$output` накапливается содержимое (код разметки) будущего HTML-документа `imageIndex.html` с галереей. В эту переменную, в частности, помещаются мини-атюры и ссылки на изображения из массива `$imageFiles`.

Добавьте в скрипт возможность отображения в галерее файлов с расширением `bmp`. Разместите в галерее произвольное изображение такого формата.

Д. Работа с форматированным текстом

9. Наберите код примера:



```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=utf-8">
5      <title>Список рассылки</title>
6  </head>
7  <body>
8      <?php
9          //проверьте наличие файла maillist.dat со списком рассылки
10         //разделитель - символ табуляции
11         $theData = file("maillist.dat");
12         foreach($theData as $line){
13             $line = rtrim($line);
14             print "<h3>$line</h3>";
15             list($name, $email) = explode("\t", $line);
16             print "<p>Name: $name</p>";
17             $message = <<<HERE
18             TO: $email:
19
20             Уважаемый $name!
21             Спасибо, что Вы - часть нашей команды. Вы просили уведомлять о новых
22             событиях.
23             Узнайте новости проекта первыми на нашем сайте!
24
25             С уважением,
26             Большой Брат,
27             всегда наблюдает за вами.
28
29             HERE;
30             print "<pre>$message</pre>";
31         } // end foreach
32     ?>
33 </body>
34 </html>

```

Сохраните его в файл `mailMerge.php`.

Здесь функция `file()` загружает содержимое файла `maillist.dat` (список рассылки) в массив. Функция `explode()` разделяет строку на поля (части, элементы) по указанному разделителю: первое поле записывает в переменную `$name`, второе – в `$email`.

Далее, list() – это не функция, а языковая конструкция PHP, позволяющая присвоить значения сразу целому списку переменных за одну операцию.

Вместо вывода результата (\$message) командой print было бы логичнее использовать mail() – функцию отправки текстового сообщения электронным письмом. Однако работа этой функции зависит от настроек на сервере, изучение которых не относится к тематике данной работы.

Создайте текстовый документ maillist.dat со списком рассылки данного письма (в каждой строке – имя контакта и адрес, разделенные табуляцией), например:

```
Дед Мороз    feedback@dom-dm.ru
Хрюша        hrusha@tvclass.ru
ПМЛ          mrysin@mail.ru
```

Проверьте работу созданного скрипта.

10. Индивидуальное задание (задача №7) на файлы (см. по [1]). Снабдите ввод и вывод подробным описанием самого задания, переменных и результата.

Литература:

1. Абрамов С.А. и др. Задачи по программированию, 1988.
2. Маклафлин Б. PHP и MySQL. Исчерпывающее руководство, 2013.
3. Суэринг С. и др. PHP 6 и MySQL 6. Библия программиста, 2010.
4. Янк К. PHP и MySQL. От новичка к профессионалу, 2013.
5. PHP: Справочник языка – Manual [Электронный ресурс]. URL: <http://php.net/manual/ru/langref.php> (дата обращения 25.03.2016).