

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

**С.В. Одиночкина**

**Web-программирование: PHP  
практикум**



**Санкт-Петербург**

УДК 004.655, 004.657, 004.62

С.В.Одиночкина

Web-программирование PHP - СПб: НИУ ИТМО, 2012. – 79 с.

В пособии излагаются методические рекомендации к выполнению лабораторных работ по дисциплине «Web-программирование PHP-технологии».

Предназначено для студентов, обучающихся по всем профилям подготовки бакалавров направления: 210700 Инфокоммуникационные технологии и системы связи.

Рекомендовано к печати Ученым советом факультета Инфокоммуникационных технологий, протокол №4 от 13 декабря 2011г.



В 2009 году Университет стал победителем многоэтапного конкурса, в результате которого определены 12 ведущих университетов России, которым присвоена категория «Национальный исследовательский университет». Министерством образования и науки Российской Федерации была утверждена программа его развития на 2009–2018 годы. В 2011 году Университет получил наименование «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

© Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, 2012

© С.В.Одиночкина, 2012.

## Лабораторная работа №9: Передача файлов на сервер

В данной лабораторной работе будут изучены основные возможности PHP для реализации передачи файлов на сервер.

### Упражнение 1: Вывод списка файлов

1. Создайте на локальном узле (в нашем случае в папке C:\WebServers\home\localhost\www\)) папки **photo** и **files** для размещения изображений и файлов соответственно. Поместите в эти папки несколько изображений и документов.
2. Создайте страницу **photo.php**, разместите на ней поясняющий текст, ниже поясняющего текста – две горизонтальные линии (между этими линиями будет выводиться список имеющихся на web-узле изображений и ссылками на эти изображения), см. *рис. 9.1*.

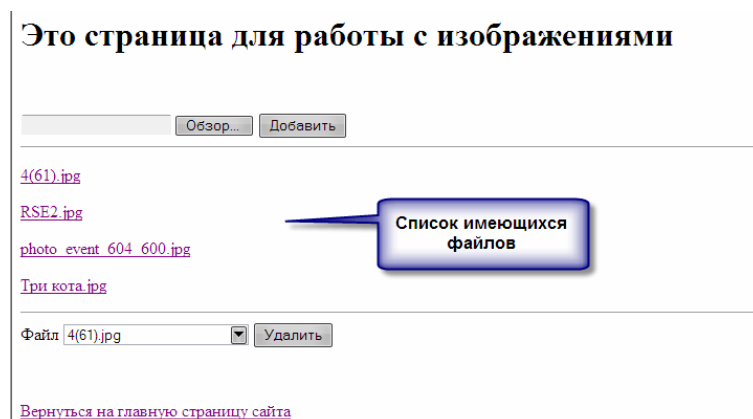


Рис. 9.1. Схема страницы *photo.php*

3. Для вывода списка файлов, уже имеющихся на сервере, необходимо создать два сценария:
  - первый будет формировать список файлов с изображениями,
  - второй - выводить этот список на страницу.
- 3.1. Формирование списка файлов. Для формирования списка файлов необходимо Получить путь к целевой папке, где хранятся необходимые файлы, а затем создать массив, в который будет помещаться необходимая информация о файлах (имена, пути к ним);

## Вариант реализации кода

```
<?php
//Получаем полный путь к папке, где хранятся графические файлы
$image_dir_path = $_SERVER['DOCUMENT_ROOT'] . "/photo";

//Запускаем просмотр папки. Функция opendir() возвращает идентификатор
//папки
$image_dir_id = opendir($image_dir_path);

//$array_files - массив, в который будут помещаться все найденные файлы
$array_files = null;

//Служебная переменная, используемая для вычисления индекса
//следующего //элемента массива $array_files

$i = 0;

//Запускаем цикл просмотра
while(($path_to_file = readdir($image_dir_id)) !== false)

//Функция readdir() возвращает полный путь к очередному файлу,
//хранящемуся //в папке, идентификатор которой был возвращен функцией
//opendir() и передан //в качестве параметра.
//$path_to_file получает полный путь к файлу для дальнейшей обработки.
//Если в папке нет непросмотренных файлов - возвращается логическое
//значение false
{
    if(($path_to_file != ".") && ($path_to_file != ".."))
    //Точки обозначают вложенные файлы: одна точка - текущая папка, две
    //точки // - папка, в которую вложена текущая папка.
    {
        $array_files[$i] = basename($path_to_file);
        $i++;
    }
    //Помещаем имя найденного файла в массив $array_files. Функция basename()
    //позволяет получить имя файла из полного пути к нему.
}
}
```

```

closedir($image_dir_id);
//closedir() удаляет из памяти переданный ей идентификатор папки, таким
//образом завершая просмотр.
?>

```

### 3.2. Вывод списка файлов на страницу.

3.1.1. Найдите код, создающий две горизонтальные линии. Создайте следующий сценарий вывода списка файлов:

```

<?php
//Получаем количество элементов массива $array_files, т.е. количество
//найденных файлов.
$array_files_count = count($array_files);
if ($array_files_count)
{
    ?>
    <hr />
    <?php
        sort($array_files);
        for ($i=0; $i<$array_files_count; $i++)
        {
            //Выводим мена хранящихся в массиве файлов на страницу
            ?>
            <p><a href="/photo/<?php echo $array_files[$i]; ?>"
                target="_blank">
                <?php echo $array_files[$i]; ?></a></p>
            <?php
                }
            ?>
            <hr />
        }
    }
}
?>

```

3.1.2. Создайте гиперссылку возврата на главную страницу и гиперссылку с главной страницы на страницу **photo.php** («Фото» в наборе гиперссылок).

4. Проверьте работу сценариев.

## Упражнение 2: Отправка файлов на сервер

Отправка файлов на web-сайт состоит из двух этапов:

- создание необходимой формы для отправки файлов на странице сайта;
- написание сценария PHP для получения отправленного посетителем сайта файла.

1. Создание формы. Стандарт HTML предусматривает т.н. **поле выбора файла**, в которое посетитель должен будет ввести имя отправляемого файла. От обычного поля ввода оно отличается тем, что позволяет работать с окном открытия файлов Windows, а также оно отправляет серверной программе не введенное в него имя файла, а сам файл. Поле выбора файла создается с помощью тэга INPUT, атрибут type которого имеет значение "file". Также в форму с данным полем необходимо добавить скрытое поле, задающее максимальный размер отправляемого файла, с именем MAX\_FILE\_SIZE. Значение этого поля (т.е. максимально возможный размер файла) определяется в байтах.

Форма, из которой будет отправляться файл, должна кодировать данные по методу multipart/form-data и передавать данные только методом POST.

```
<!-- Форма для отправки файла на сервер -->
<form name = "file_upload" action="photo.php"
      enctype="multipart/form-data" method="post">
  <input type="hidden" name="MAX_FILE_SIZE" value="65536" />
  <input type="file" name="file_upload" />
  <input type="submit" name="submit" value="Добавить" />
</form>
```

В данном случае максимальный размер файла равен 65536 байта, т.е. 64 Кбайта. Данный размер при необходимости может быть увеличен.

2. Получение отправленного файла. Все принятые файлы помещаются интерпретатором PHP в особую служебную папку, которая не является частью сайта (т.н. «буферная» папка). Сведения обо всех принятых и помещенных в буферную папку файлах хранятся во встроенном массиве PHP \$\_FILES. Каждый элемент массива соответствует принятому файлу и представляет собой вложенный массив, содержащий различные сведения о файле. В начало страницы поместите следующий код:

```
<?php
//Сценарий отправки файла на сервер
//Проверяем, была ли выполнена отправка файла. Далее реализуем
сценарий.
if (isset($_POST["MAX_FILE_SIZE"]))
{
```

```

$tmp_file_name = $_FILES["file_upload"]["tmp_name"];
$dest_file_name = $_SERVER['DOCUMENT_ROOT'].
    "/photo/".$_FILES["file_upload"]["name"];
move_uploaded_file($tmp_file_name, $dest_file_name);
}
?>

```

Функция ***move\_uploaded\_file ( string filename, string destination )*** проверяет, является ли файл *filename* загруженным на сервер (переданным по протоколу HTTP POST). Если файл действительно загружен на сервер, он будет перемещён в место, указанное в аргументе *destination*.

### Упражнение 3: Удаление файла с сервера

Для удаления файла надо выбрать необходимый файл. Это делается с помощью формы, в которую помещается список для выбора удаляемого файла из имеющихся на сервере файлов. Для заполнения этого списка необходимо использовать уже созданный массив `$array_files`.

Реализация удаления происходит в два этапа:

- Создание формы для удаления файла;
- Создание сценария удаления файла.

#### 1. Создание формы для выбора удаляемого файла

```

<!-- Форма для удаления файла с сервера -->
<form name="file_delete" action="photo.php" method="post"

enctype="application/x-www-form-urlencoded">
Файл <select name = "file_delete" size="1">

    <option><option></select>
<input type="submit" name="submit" value="Удалить" />
</form>

```

Парные тэги `SELECT` создает список. Пункты списка создаются тэгами `OPTION`. Сам список имеющихся на сервере файлов необходимо получить из созданного ранее массива `$array_files`.

```

<!-- Форма для удаления файла с сервера -->
<form name="file_delete" action="photo.php" method="post"

```

```

enctype="application/x-www-form-urlencoded">
Файл <select name = "file_delete" size="1">
<?php for ($i=0; $i<$array_files_count; $i++) { ?>
<option><?php echo $array_files[$i]; ?></option>
<?php } ?></select>
<input type="submit" name="submit" value="Удалить" />
</form>

```

Во включенном в форму цикле создается столько пунктов (тэгов <option>), сколько элементов присутствует в массиве \$array\_files.

## 2. Создание сценария удаления файла.

```

<?php
//Сценарий удаления файла
//Сначала проверяем, было ли запущено удаление файла
if (isset($_POST["file_delete"]))
{
    //Формируем полное имя файла
    $file_name = $_SERVER['DOCUMENT_ROOT'] . "/photo/".
        $_POST["file_delete"];

    //Функция unlink() удаляет файл
    unlink($file_name);
}
?>

```

## 3. Сохраните изменения, проверьте корректность работы.



#### Упражнение 4: Создание страницы работы с файлами

Самостоятельно создайте страницу для управления прочими файлами **files.php**. Она во всем аналогична странице **photo.php**, кроме:

1. Она должна обрабатывать файлы, находящиеся в папке `files`.
2. Просмотр файлов делать необязательно (это не изображения), следовательно не надо делать гиперссылки из имени каждого файла.
3. Необходимо увеличить максимальный размер файла хотя бы до 1 Мбайта.
4. Сделайте гиперссылки, связывающие данную страницу с главной страницей.