

# Загрузка файлов на сервер

## Техническое задание

Создать форму для загрузки файлов на сервер в отдельную папку

### Новое

`$_FILES` — Переменные файлов, загруженных по HTTP. Ассоциативный массив (array) элементов, загруженных в текущий скрипт через метод HTTP POST.

`move_uploaded_file()` – функция, для сохранения временного файла, которая переместит его из временной папки в нужную нам папку. Первым аргументом она принимает путь до временного файла (в нашем случае ниже - это `$_FILES['attachment']['tmp_name']`), а вторым аргументом – путь, по которому нужно его сохранить. Функция вернёт `true`, если всё ок, и `false`, если что-то пошло не так.

`Pathinfo` – функция файловой системы, возвращает информацию о пути к файлу. `pathinfo()` возвращает информацию о `path` в виде ассоциативного массива или строки, в зависимости от `flags` (`PATHINFO_DIRNAME`, `PATHINFO_BASENAME`, `PATHINFO_EXTENSION` и `PATHINFO_FILENAME`).

`UPLOAD_ERR_OK` - Значение: 0; Ошибок не возникло, файл был успешно загружен на сервер. (PHP возвращает код ошибки наряду с другими атрибутами принятого файла. Он расположен в массиве, создаваемом PHP при загрузке файла, и может быть получен при обращении по ключу `error`.)

`file_exists` — функция файловой системы, проверяет существование указанного файла или каталога. Возвращает `true` или `false`.

### Реализация

1. Сначала создаем файл `upload.php`. В нём создаем форму

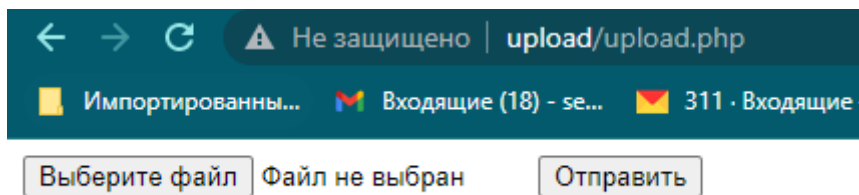
```
2. <html>
3. <head>
4.     <title>Загрузка файла</title>
5. </head>
6. <body>
7. <form action="/upload.php" method="post" enctype="multipart/form-data">
8.     <input type="file" name="attachment">
9.     <input type="submit">
10.</form>
11.</body>
12.</html>
```

- 1.1 В форме в `action` помещаем адрес к этому же файлу, добавляем метод `POST`, способ кодирования определяем как `multipart/form-data` (значит «данные не кодируются. Это значение применяется при отправке файлов»).
- 1.2 В теге `input` в атрибуте `type` ставим тип «file», а название `attachment`. Добавляем `input` с типом `submit`.
- 1.3 Так как мы указали имя `attachment`, то узнать информацию о загружаемом нами файле можно получить по соответствующему ключу - `$_FILES['attachment']`.

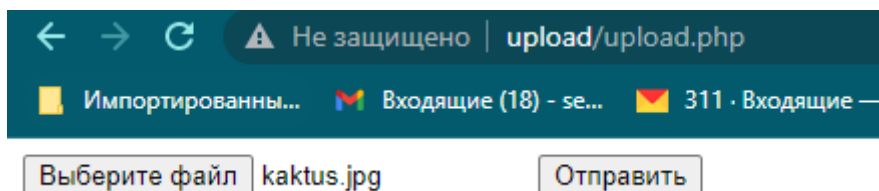
1.4 Добавим проверку, чтобы посмотреть как загружаются в файл. Если массив \$\_FILES не пустой, то выводим содержимое массива (if (!empty(\$\_FILES))).

```
2 <?php
3 if (!empty($_FILES)) {
4     var_dump($_FILES);
5 }
6 ?>
7 <html>
8 <head>
9     <title>Загрузка файла</title>
10 </head>
11 <body>
12 <form action="/upload.php" method="post" enctype="multipart/form-data">
13     <input type="file" name="attachment">
14     <input type="submit">
15 </form>
16 </body>
17 </html>
```

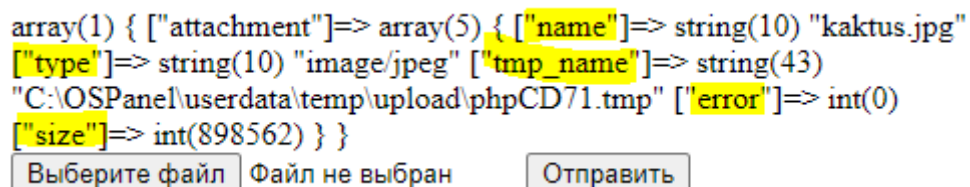
1.5. Запускаем нашу страницу upload.php



1.6 Выбираем какой-нибудь файл



1.7 Нажимаем отправить



1.7.1 Выведен массив:

- name – имя файла, переданное из браузера; //kaktus.jpg
- type – тип файла, в моём случае это image/png; //image/jpeg
- tmp\_name – путь до временного файла, который загрузился на сервер, но если с ним до конца запроса ничего не сделать, то он удалится;
- error – код ошибки при загрузке файла, если 0 – то ошибки нет; //0

- *size* – размер загруженного файла в байтах.//`int(898562)`.

2. Файл загрузился - проверка пройдена. Удаляем (if (!empty(\$\_FILES)).

2.1 Сперва создаем условие, если в переданном массиве \$\_FILES есть элемент под названием attachment, который также является массивом и он не пустой, то создаем переменную \$file, куда помещаем содержимое элемента attachment (file = \$\_FILES['attachment'];).

2.2 Далее создаем переменную \$srcFileName, в которую помещаем значение ключа name в массиве (\$srcFileName = \$file['name'];). То есть в переменной будет находиться имя загружаемого файла.

2.3 Далее создаем переменную \$newFilePath, в которую с помощью \_\_DIR\_\_ помещаем путь '/uploads/' - папка с нашими загруженными файлами и с помощью конкатенации склеиваем с переменной \$srcFileName (с именем файла).

```
<?php
if (!empty($_FILES['attachment'])) {
    $file = $_FILES['attachment'];

    $srcFileName = $file['name'];
    $newFilePath = __DIR__ . '/uploads/' . $srcFileName;

    $allowedExtensions = ['jpg', 'png', 'gif'];
    $extension = pathinfo($srcFileName, PATHINFO_EXTENSION);
    if (!in_array($extension, $allowedExtensions)) {
        $error = 'Загрузка файлов с таким расширением запрещена!';
    } elseif ($file['error'] !== UPLOAD_ERR_OK) {
        $error = 'Ошибка при загрузке файла.';
    } elseif (file_exists($newFilePath)) {
        $error = 'Файл с таким именем уже существует';
    } elseif (!move_uploaded_file($file['tmp_name'], $newFilePath)) {
        $error = 'Ошибка при загрузке файла';
    } else {
        $result = 'http://myproject.loc/uploads/' . $srcFileName;
    }
}
?>
<html>
<head>
    <title>Загрузка файла</title>
</head>
<body>
<?php if (!empty($error)): ?>
    <?= $error ?>
<?php elseif (!empty($result)): ?>
    <?= $result ?>
<?php endif; ?>
<br>
```

```
<form action="/upload.php" method="post" enctype="multipart/form-data">
  <input type="file" name="attachment">
  <input type="submit">
</form>
</body>
</html>
```

2.4 Создаем переменную \$allowedExtensions, в которую помещаем массив из форматов файлов, которые мы разрешаем пользователю загружать на сервер ( \$allowedExtensions = ['jpg', 'png', 'gif'];).

2.5 Создаем переменную с именем расширения \$extension, с помощью которой мы получим расширение загружаемого пользователем файла.

2.5.1 В значение переменной \$extension помещаем функцию pathinfo. Первым параметром указываем имя файла \$srcFileName (в котором у нас находится имя загружаемого файла, а вторым параметром указываем константу PATHINFO\_EXTENSION (таки образом мы говорим функции, чтобы вернулось именно расширение файла).

2.6. Создается блок дополнительных условий.

2.6.1 Если: в массиве \$allowedExtensions (который с разрешенными типами расширений) нет извлеченного из загружаемого пользователем файла расширения (\$extension), то создаем переменную \$error со значением 'Загрузка файлов с таким расширением запрещена!'.

2.6.2 Если: здесь мы возвращаемся к массиву \$file. Если у ключа ['error'] значение не равно UPLOAD\_ERR\_OK, а это значит, что файл не был успешно загружен, то создаем переменную \$error со значением 'Ошибка при загрузке файла.'.

2.6.3 Если: с помощью функции file\_exists проверяем существует ли такой файл (file\_exists(\$newFilePath)).

2.6.4 Функция move\_uploaded\_file() при успешной работе возвращает true, иначе false. Следующее условие: функция move\_uploaded\_file() принимает временный файл (если файл найден и всё ок), то вторым аргументом говорит по какому пути нужно сохранить файл (второй аргумент – это наша переменная \$newFilePath, в которой находится путь). Если move\_uploaded\_file() возвращает false, то создаем переменную \$error со значением 'Ошибка при загрузке файла'.

2.6.5 Иначе: создаем переменную \$result, в которую помещаем путь (куда поместить загружаемый файл) и имя файла ('http://myproject.loc/uploads/' . \$srcFileName;).

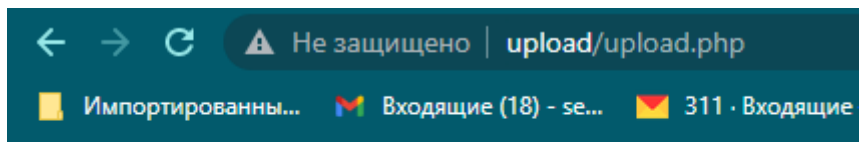
2.7 Добавляем проверку на наличие ошибок и вывод результата. В теле тега body помещаем следующие условия:

2.7.1 Если переменная \$error не пуста, то выводим значение переменной \$error.

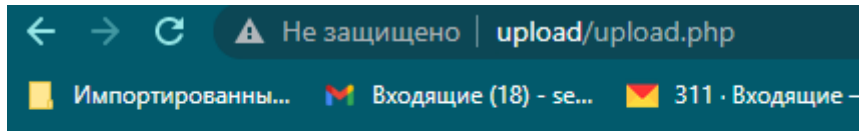
2.7.2 Если переменная \$result не пуста, то выводим значение переменной \$result. Условие закрывается.

## Тестирование

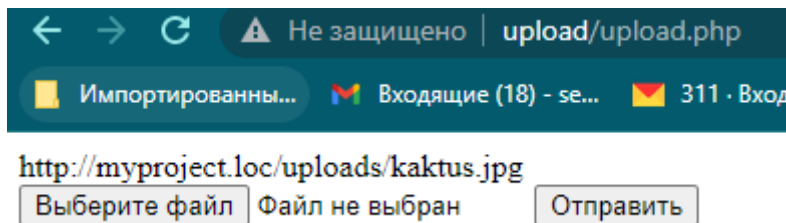
1. Заходим на нашу страницу upload.php.



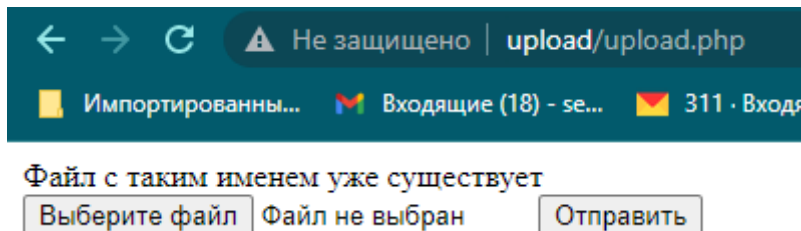
2. Выбираем файл с разрешенными разрешениями ('jpg', 'png', 'gif').



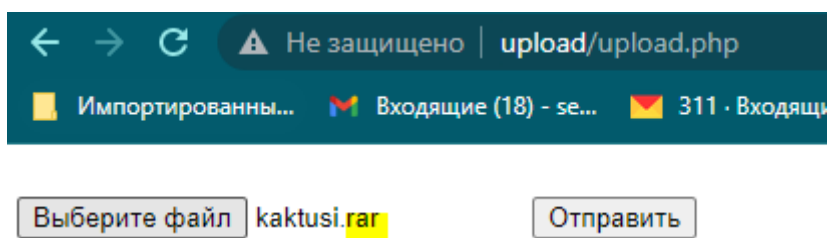
3. Нажимаем отправить. Будет выведен путь к файлу.



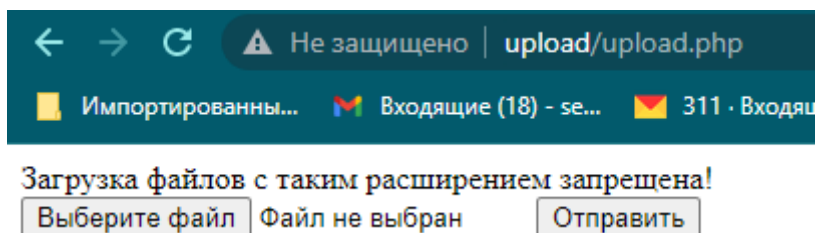
4. Выберем еще раз этот же файл. Будет выведена ошибка



5. Выберем файл с неразрешенным расширением



6. Будет выведена ошибка



7. Программа успешно работает.