Determine which superglobal function can be used to send files from client browsers to the web server? Describe the process on how you can send the files with practical examples and explain what is the first thing you should do as a web developer to handle the file on the server?

Make sure to include screenshot and source code of the designed “uploadfile.php” page.

**Answer:**

PHP uses special predefined associative arrays called superglobal variables that allow the programmer to easily access HTTP headers, query string parameters, and other commonly needed information.

They are called superglobal because they are always in scope, and always defined.

The $\_GET and $\_POST arrays are the most important superglobal variables in PHP since they allow the programmer to access data sent by the client in a query string.

The PHP superglobal variables are:

1. $GLOBALS
2. $\_SERVER
3. $\_REQUEST
4. $\_POST
5. $\_GET
6. $\_FILES
7. $\_ENV
8. $\_COOKIE
9. $\_SESSION

The PHP superglobal functions used to upload the files from Client Browser to the web server using PHP is: $\_FILES.

Global PHP variable called **$\_FILES**. This variable is an associate double dimension array and keeps all the information related to uploaded file. So if the value assigned to the input's name attribute in uploading form was **file**, then PHP would create following five variables

Superglobal Description

$\_FILES['my-file']['name'] Original Name of File Before It Was Uploaded

$\_FILES['my-file']['type'] The MIME Type of File, Provided By the Browser

$\_FILES['my-file']['size'] Size of the File (In Bytes)

$\_FILES['my-file']['tmp\_name'] Location of Temporary File on Server

$\_FILES['my-file']['error'] Any Error Codes Resulting From the File Upload

PHP allow the user to select files to upload into a user specified directory. The user needs to send the details for the file through HTML forms.

These details are processed in the PHP Script by using the superglobal variable named $\_FILES. File upload can be implemented by following the below steps.

1. Create HTML form with the required element to choose a file. Define as enctype="multipart/form-data"
2. Specify the directory path as a target to the uploaded file.
3. Check whether $\_FILES global array not empty.
4. Ensure that the selected file is available in the temporary directory
5. And then, call a function to upload a file from temporary location to the user specified target.

**HTML required for file uploads:**

1. You must ensure that the HTML form uses the HTTP POST method, since transmitting a file through the URL is not possible.
2. You must add the enctype="multipart/form-data" attribute to the HTML form that is performing the upload so that the HTTP request can
3. You must include an input type of file in your form.

**Handling file upload in php:**

The $\_FILES array will contain a key=value pair for each file uploaded in the post.

The key for each element will be the name attribute from the HTML form, while the value will be an array containing information about the file as well as the file itself.

The keys in that array are the name, type, tmp\_name, error, and size.

* **name** is a string containing the full file name used on the client machine, including any file extension.
* **type** defines the MIME type of the file
* **tmp\_name** is the full path to the location on your server where the file is being temporarily stored.
* **error** is an integer that encodes many possible errors and is set to UPLOAD\_ERR\_OK (integer value 0) if the file was uploaded successfully.

**size** is an integer representing the size in bytes of the uploaded file

**Source code:**

**Client side script code:**

<!DOCTYPE html>

<head>

<style type="text/css">

form{

padding-top: 50px;

padding-bottom: 50px;

text-align: center;

border: 1px solid black;

margin-right: 200px;

margin-left: 200px;

}

input[type=submit]{

margin-top: 10px;

}

</style>

</head>

<html>

<body>

<form action="upload.php" method="post" enctype="multipart/form-data">

Select image to upload:

<input type="file" name="fileToUpload" id="fileToUpload"><br>

<input type="submit" value="Upload Image" name="submit">

</form>

</body>

</html>

**Server side script code:**

<?php

$target\_dir = "/Applications/XAMPP/xamppfiles/htdocs/mor/images/";

$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);

$uploadOk = 1;

$imageFileType = pathinfo($target\_file,PATHINFO\_EXTENSION);

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;

}

}

if (file\_exists($target\_file)) {

echo "Sorry, file already exists.";

$uploadOk = 0;

}

if ($\_FILES["fileToUpload"]["size"] > 500000) {

echo "Sorry, your file is too large.";

$uploadOk = 0;

}

if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"

&& $imageFileType != "gif" ) {

echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";

$uploadOk = 0;

}

if ($uploadOk == 0) {

echo "Sorry, your file was not uploaded.";

} 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.";

}

}

?>

**Screen shots:**



















