

# File Upload in PHP - PHP File Upload Script

Authored by: **Administrator** [[help@knowledgebase-script.com](mailto:help@knowledgebase-script.com)]

Saved From: <http://www.articlediary.com/article/file-upload-in-php-php-file-upload-script-34.html>

---

This script will help you learn how to **upload files with PHP**. PHP is undoubtedly the best programming language when it comes to web programming. It gives us so many features and capabilities, a few of which we've discussed already. So continuing with that, today we will see how we can use a few lines of PHP code to create a **File Upload Script** that would allow file uploading right from the web browser. File upload feature is definitely useful kinds of website but at the same time very much vulnerable to malicious attacks as well. So use it with a lot of precautions!

For this example, we will need a front-end web page (upload.html) that'll accept the file from the user and a backend PHP file upload script (file-upload.php) to process and store the file. Let us look at the codes of each file.

```
<html>
<body>
<form action="file-upload.php" method="post"
enctype="multipart/form-data">
<label for="file">Filename:</label>
<input type="file" name="file" id="file" />
<br />
<input type="submit" name="submit" value="Submit" />
</form>
</body>
</html>
```

Here we have a HTML form that calls the script on submission. The method of data sending should be `POST` and there should be an `enctype` as `multipart/form-data` which means we can upload binary form data. The input type `file` opens the File Input Box that'd let the user browse for the file. The submit button `Upload` submits the form as usual.

PHP File Upload Script (file-upload.php)

```
<?php
if ($_FILES["file"]["error"] > 0)
{
echo "Error:: " . $_FILES["file"]["error"] . "<br />";
}
else
{
if (file_exists($_FILES["file"]["name"]))
{
echo $_FILES["file"]["name"] . " already exists. ";
}
else
{
move_uploaded_file($_FILES["file"]["tmp_name"], $_FILES["file"]["name"]);
echo "File <b>" . $_FILES["file"]["name"] . "</b> uploaded successfully!";
}
}
}
```

?>

Just like we had POST form data in `$_POST[]` and GET data in `$_GET[]` array same way to have files sent to script we use `$_FILES[]` array.

1. `$_FILES["file"]["error"]`: Tells us if there is any error in uploading. Note that "file" the first string index of the array is the name of the "file" input type from the HTML form.
2. `$_FILES["file"]["tmp_name"]`: All the files sent to any script is stores in a temporary directory by PHP, this tells us the location of that temporary file.
3. `$_FILES["file"]["name"]`: It tells us the name of the file on the users" computer.

**move\_uploaded\_file()**: As files sent to scripts are just stored temporarily, we have to save it to some place, for this we use this PHP function. It's first parameter is the temporary file location and second is the place we need to store the file and with what name. We are storing it to the same directory the script is in and with name same as on the users" computer.

**NOTE: This is just meant for example purpose and you shouldn't have this on your server as it is an open invitation to hackers/spammers. If you need upload facility on your website in all the cases you should have some kind of authentication system that'd only allow registered users to upload anything. you should also accept only certain file types. you wouldn't like someone uploading/running spamming script off your server. Would you? there may be a few other precautions that you may need to take depending on the purpose you intend to use the script for.**