HOME   SUBSCRIBE!   NEWSLETTER   WRITE FOR US   PRIVACY   TERMS   🔍

php

How to Download a File in PHP

7 months ago • by Fahmida Yesmin

Generally, no PHP script is required to download a file with the extensions **exe** and **zip**. If the file location of this type of file is set in the **href** attribute of the anchor element, then the file

downloaded forcibly in PHP using **the readfile()** function that does not download automatically. This tutorial shows you how to forcibly download any file using PHP script.
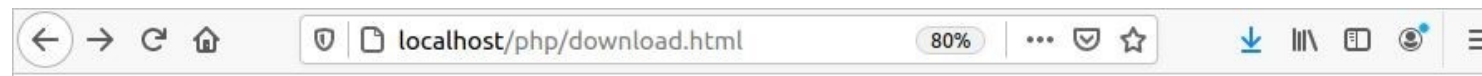
Check Download Links

It was previously mentioned that **zip** and **exe** files download automatically, without using PHP script. First, create an HTML file with the following code. Here, the four anchor elements are defined to download the four types of files. These file types include TEXT, ZIP, PDF, and JPG files.

Download.html

```html
<html>
<head>
<title>Download Files</title>
</head>
<body>
<p><a href="abc.txt">Download TEXT file</a></p>
<p><a href="horizon.zip">Download ZIP file</a></p>
<p><a href="lecture.pdf">Download PDF file</a></p>
<p><a href="rose.jpg">Download JPG file</a></p>
</body>
</html>
```
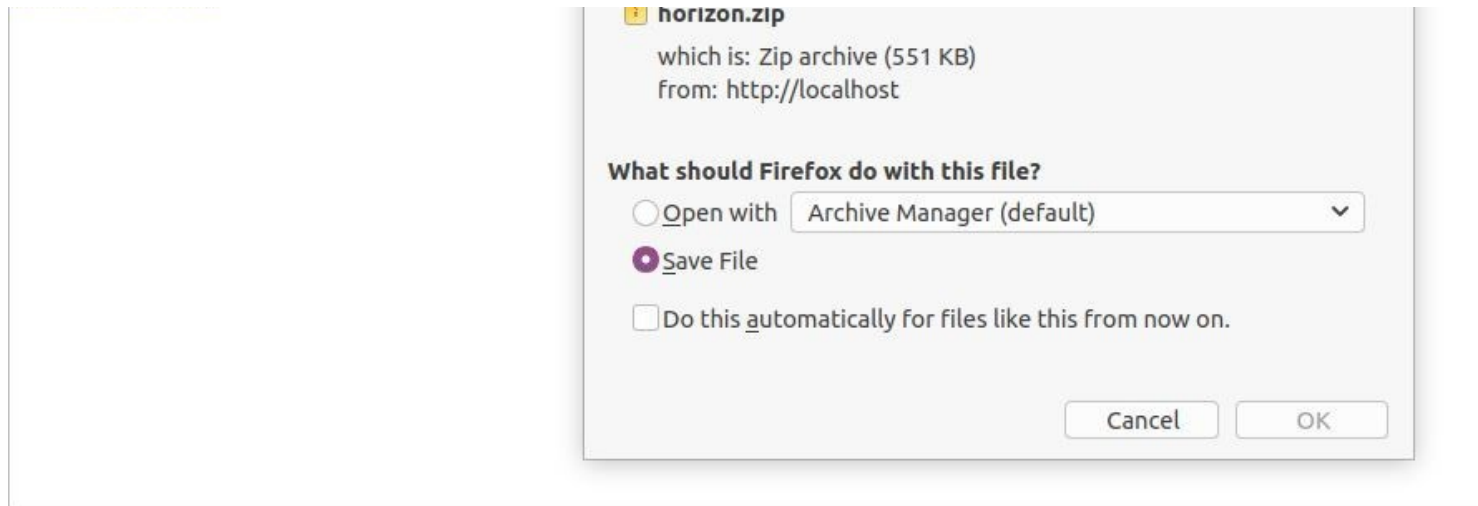
Output

The following dialog box will appear to download the file after clicking the zip file link. The user can then download the file or open the file in the archive manager.

If you click on the image file, the image will be opened automatically in the browser, as shown in the following output. You must save the file to make a copy of the image file in the local drive. In the same way, when you click on PDF and TEXT file links, the content of the file will be opened in the browser without downloading the file. The solution to this problem is to download the file forcibly using the built-in PHP **readfile()** function.

MY LATEST VIDEOS

Top Bash Tricks and Tips to be more effective on th…

Top Bash Tricks and Tips to be more effective on the shell

Download File Using readfile() Function

The **readfile()** function is used in PHP script to forcibly download any file of the current location, or the file with the file path. The syntax of this function is given below.

Syntax

int readfile (string $filename [, bool $use_include_path = false [, resource $context]])

This function can take three arguments. The first argument is mandatory, and the other two arguments are optional. The first argument, **$filename**, stores the filename or filename with the path that will download. The default value of the second parameter, **$use_include_path**, is false and will be set to true if the filename with the path is used in the first argument. The third argument, **$context**, is used to indicate the context stream resource. This function returns the number of bytes read from the file mentioned in the first argument. The uses of this function are shown in the following two examples.

Example 1: Download File with Filename

In this example, we will create an HTML file with the following code, where the file name will be passed as a parameter of the URL named **path**, and the value of this parameter will be passed to the PHP file named **download.php**.

download2.html

```html
<html>
<head>
<title>Download Files</title>
</head>
<body>
<p><a href="download.php?path=abc.txt">Download TEXT file</a></p>
<p><a href="download.php?path=horizon.zip">Download ZIP file</a></p>
<p><a href="download.php?path=lecture.pdf">Download PDF file</a></p>
<p><a href="download.php?path=rose.jpg">Download JPG file</a></p>
</body>
```

defined, the **file_exists()** function is used to check whether the file exists in the server. Next, the **header()** function is used to set the necessary header information before using the **readfile()** function. The **basename()** function is used to retrieve the filename, and the **filesize()** function is used to read the size of the file in bytes, which will be shown in the opening dialog box to download the file. The **flush()** function is used to clear the output buffer. The **readfile()** function is used with the filename only, here.

download.php

```php
<?php

if(isset($_GET['path']))
{
//Read the filename
$filename = $_GET['path'];
//Check the file exists or not
if(file_exists($filename)) {

//Define header information
header('Content-Description: File Transfer');
header('Content-Type: application/octet-stream');
header("Cache-Control: no-cache, must-revalidate");
header("Expires: 0");
header('Content-Disposition: attachment; filename="'.basename($filename).'"');
header('Content-Length: ' . filesize($filename));
header('Pragma: public');

//Clear system output buffer
flush();

//Read the size of the file
readfile($filename);

//Terminate from the script
die();
}
```

?>

Output

The following output will appear after clicking the download link of the image file. The file size of the **rose.jpg** image is **27.2 KB**, as shown in the dialog box. You can download the file by selecting the **Save File** radio button and pressing the **OK** button.

Example 2: Download File with File Path

If the file exists at the given file location, the file path will be required to mention in the URL. In this example, we will create an HTML file with the following code, which will pass the filename with the file path:

download3.html

```html
<html>
<head>
<title>Download Files</title>
</head>
<body>
<p><a href="download.php?path=downloads/lecture.pdf">Download PDF file</a></p>
<p><a href="download2.php?path=downloads/rose.jpg">Download JPG file</a></p>
</body>
</html>
```

We will create a PHP file with the following code to download a file from the file path. The

Two arguments are used in the **readfile()** function.

download2.php

```php
<?php
if(isset($_GET['path']))
{
//Read the url
$url = $_GET['path'];

//Clear the cache
clearstatcache();

//Check the file path exists or not
if(file_exists($url)) {

//Define header information
header('Content-Description: File Transfer');
header('Content-Type: application/octet-stream');
header('Content-Disposition: attachment; filename="'.basename($url).'"');
header('Content-Length: ' . filesize($url));
header('Pragma: public');

//Clear system output buffer
flush();

//Read the size of the file
readfile($url,true);

//Terminate from the script
die();
}
else{
echo "File path does not exist.";
}
}
echo "File path is not defined."

?>
```

Output

download file using php

▶

Conclusion

This article provided a simple way to forcibly download any file using the PHP script, to help readers to add the download feature in their script.

Fahmida Yesmin

I am a trainer of web programming courses. I like to write article or tutorial on various IT topics. I have a YouTube channel where many types of tutorials based on Ubuntu, Windows, Word, Excel, WordPress, Magento, Laravel etc. are published: Tutorials4u Help.

View all posts

RELATED LINUX HINT POSTS

Use of usleep() function in PHP

Write into a File in PHP using fwrite()

Use of usleep() Function in PHP

Write into a file in PHP using fwrite()

Change the string into uppercase in PHP

Generate a random number in PHP

How to install Xdebug and use it in PHP on Ubuntu?

AN ELITE CAFEMEDIA PUBLISHER