

# PHP – Create Directory

Computer files are stored in the local storage device (called **drive**) in a hierarchical order, where a directory contains one or more files as well as subdirectories. Respective DOS commands defined in operating systems Windows, Linux etc. are used to create and manage directories.

PHP provides directory management functions to create a directory, change the current directory and remove a certain directory.

This chapter discusses the usage of the following directory functions in PHP –

## The mkdir() Function

The mkdir() function creates a new directory whose path is given as one of the parameters to the function

```
mkdir(  
    string $directory,  
    int $permissions = 0777,  
    bool $recursive = false,  
    ?resource $context = null  
): bool
```

## Parameters

- **\$directory** – The first parameter \$directory is mandatory. It is a string with either absolute or relative path of the new directory to be created.
- **\$permissions** – The second parameter \$permissions is an octal number with four octal digits. The first digit is always zero, second specifies permissions for the owner, third for the owner's user group and fourth for everybody else.

Each digit is the sum of values for each type of permission –

- 1 = execute permission
- 2 = write permission
- 4 = read permission



The default value of \$permissions parameters is **0777**, which means the directory is created with execute, write and read permissions enabled.

Note that the \$permissions parameter is ignored when working on Windows OS.

- **\$recursive** – If true, then any parent directories to the directory specified will also be created, with the same permissions.
- **\$context** – This optional parameter is the stream resource.

The mkdir() function returns either true or false, indicating if the function has been successfully executed or not.

## Examples

Here are some examples of mkdir() function.

The following call to mkdir() creates a subdirectory inside the current working directory. The dot indicates that the path is relative.

```
$dir = "./mydir/";  
mkdir($dir);
```

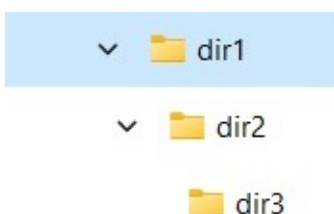
We can give the string parameter that contains the absolute path of the directory to be created.

```
$dir = "c:/newdir/";  
mkdir($dir);
```

The following call to mkdir() contains nested directory structure inside the current directory, as the \$recursive parameter is set to true.

```
$dirs = "./dir1/dir2/dir3/";  
mkdir($dirs, 0777, true);
```

The Windows explorer will show the nested directory structure as follows –



## The chdir() Function

The `chdir()` function in PHP corresponds to the **`chdir`** or **`cd`** command in Linux/Windows. It causes the current directory to be changed as required.

```
chdir(string $directory): bool
```

The string parameter to this function is either an absolute or relative path of a directory to which the current directory needs to be changed to. It returns true or false.

Explore our [latest online courses](#) and learn new skills at your own pace. Enroll and become a certified expert to boost your career.

## The `getcwd()` Function

The `getcwd()` function works similar to `pwd` command in Ubuntu Linux, and returns the path to the current working directory.

### Example

With the following code snippet, PHP displays the current working directory before and after changing the current working directory. A couple of files are created inside the new current directory. With the `scandir()` function, the files are listed.

```
<?php
    echo "current directory: " . getcwd() . PHP_EOL;
    $dir = "./mydir";
    chdir($dir);
    echo "current directory changed to: " . getcwd() . PHP_EOL;

    $fp = fopen("a.txt", "w");
    fwrite($fp, "Hello World");
    fclose($fp);

    copy("a.txt", "b.txt");
    $dir = getcwd();
    foreach(scandir($dir) as $file)
        echo $file . PHP_EOL;
?>
```

It will produce the following **output** –

```
current directory: C:\xampp\php
current directory changed to: C:\xampp\php\mydir
```

```
.  
..  
a.txt  
b.txt
```

## The rmdir() Function

The rmdir() function removes a certain directory whose path is given as parameter. The directory to be removed must be empty.

```
$dir = "c:/newdir/";  
rmdir($dir) or die("The directory is not present or not empty");
```