

# PHP – Listing Files

Windows command DIR and Linux command ls both display the list of files in the current directory. These commands can be operated with different switches to apply conditions on the list of files displayed. PHP provides a couple of options for programmatically listing files in a given directory.

## The readdir() Function

The opendir() function in PHP is similar to fopen() function. It returns handles to the directory so that the contents of the directory can be read from in a serialized manner.

```
opendir(string $directory, ?resource $context = null): resource|false
```

This function opens up a directory handle to be used in the subsequent closedir(), readdir(), and rewinddir() calls.

The **readdir()** function reads the next available entry from the stream handle returned by **opendir()** function.

```
readdir(?resource $dir_handle = null): string|false
```

Here, **dir\_handle** is the directory handle previously opened with opendir().not specified, the last link opened by opendir() is assumed.

The **closedir()** function is similar to **fclose()** function. It closes the directory handle.

```
closedir(?resource $dir_handle = null): void
```

The function closes the directory stream indicated by dir\_handle. The stream must have previously been opened by opendir().

## Example

The following PHP code reads one file at a time from the currently logged directory.

```
<?php
    $dir = getcwd();

    // Open a known directory, and proceed to read its contents
    if (is_dir($dir)) {
```



```
if ($dh = opendir($dir)) {  
    while (($file = readdir($dh)) !== false) {  
        echo "filename:" . $file . "\n";  
    }  
    closedir($dh);  
}  
}
```

## The scandir() Function

The scandir() function retrieves the files and subdirectories inside a given directory.

```
scandir(string $directory,  
int $sorting_order = SCANDIR_SORT_ASCENDING,  
?resource $context = null): array|false
```

The "sorting\_order" by default is alphabetical in ascending order. If this optional parameter is set to SCANDIR\_SORT\_DESCENDING, then the sort order becomes alphabetical in descending order. If it is set to SCANDIR\_SORT\_NONE, then the result becomes unsorted.

## Example

With the following PHP code, the scandir() function returns an array of files in the given directory.

```
<?php  
$dir = "c:/xampp/php/mydir/";  
  
$files = scandir($dir);  
var_dump($files);  
?>
```

It will produce the following **output** –

```
array(4) {  
    [0]=>  
    string(1) "."  
    [1]=>  
    string(2) ".."  
    [2]=>  
    string(5) "a.txt"
```

```
[3]=>
string(5) "b.txt"
}
```

You can use a **foreach** loop to traverse the array returned by the `scandir()` function.

```
<?php
    $dir = "c:/xampp/php/mydir/";

    $files = scandir($dir);
    foreach ($files as $file)
        echo $file . PHP_EOL;
?>
```

It will produce the following **output** –

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