Instructions for using find command in Linux

The find command is one of the most important and handy commands on a Linux system. As its name suggests, the command can find files on a Linux PC based on a variety of conditions and variables you set.

You can search for files by permission, user, group, file type, date, size, and other possible criteria using the find command.

The find command is available on most Linux distributions by default, so you don't need to install packages for it.

In this guide, **Macworld** will show you how to find files on Linux using the combination of the various popular search expression in the command line.

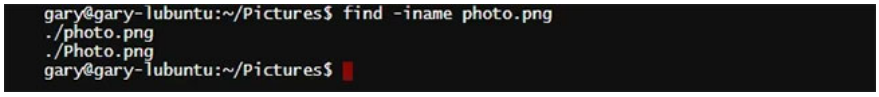
Search file by name in current directory

The most obvious way to search for files is by name. To find a file by name in the current directory, run:

```
find . -name photo.png
```

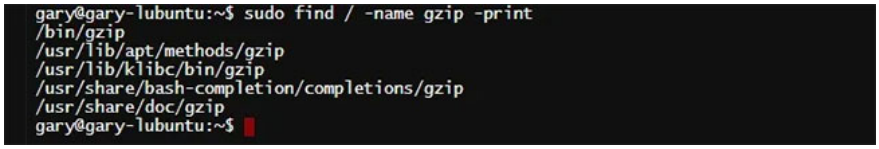
If you want to find a file by name that has both upper and lower case letters, run:

```
find . -iname photo.png
```



If you want to find a file in the root directory, put the **sudo** prefix before your search. This will give you all the necessary permissions to do so and also has a '/' icon that allows Linux to search in the root directory. Finally, the **-print** expression displays the directories in the search results. If you're looking for Gzip, you must type:

```
sudo find / -name gzip -print
```



Find files in specific directory

If you want to find files in a specific directory like '**/ home**' , run:

```
find /home -name filename.txt
```

If you want to find files with the '**.txt**' extension in the '**/ home**' **directory** , run:

```
find /home -name *.txt
```

To find files named '**test.txt**' in multiple directories like '**/ home**' and '**/ opt**' run:

```
find /home /opt -name test.txt
```

To find files hidden in the '**/ home**' **directory** , run:

```
find /home -name ".*"
```

To find a single file named '**test.txt**' and delete it, run:

```
find /home -type f -name test.txt -exec rm -f {}
```

To find all empty files in the '**/ opt**' **directory** , run:

```
find /opt -type f -empty
```

Find folder by name

If you want to find all folders named '**testdir**' in the '**/ home**' **directory** run:

```
find /home -type d -name testdir
```

To archive all empty directories under '**/ home**' run:

```
find /home -type d -empty
```

Find files with certain permissions

The find command can be used to find files with specific permissions using the **perm** option **.**

To find all files with permission as '**777**' in the '**/ home**' **directory** , run:

```
find /home -type f -perm 0777 -print
```

To find all files that do not have "**777**" permission , run:

```
find . -type f ! -perm 777
```

To find all read-only files, run:

```
find /home -perm /u=r
```

To find all executable files, run:

```
find /home -perm /a=x
```

To find all Sticky Bit file sets that have permission to '**553**' , run:

```
find /home -perm 1553
```

To find all SUID file sets, run:

```
find /home -perm /u=s
```

To find all files with permissions as '**777**' and change their permissions to '**700**' run:

```
find /home -type f -perm 0777 -print -exec chmod 700 {} ;
```

Finds files and folders based on date and time

To find all files in '**/ opt**' that were modified 20 days ago run:

```
find /opt -mtime 20
```

To find all files under '**/ opt**' accessed 20 days ago run:

```
find /opt -atime 20
```

To find all files under modified '**/ opt**' for 30 to 50 days run:

```
find /opt -mtime +30 -mtime -50
```

To find all files under '**/ opt**' changed in the last two hours run:

```
find /opt -cmin -120
```

Find files and folders based on size

To find all 10MB files in the '**/ home' directory** run:

```
find /home -size 10M
```

To find all files in '**/ home' directory** larger than 10MB and less than 50MB, run:

```
find /home -size +10M -size -50M
```

To find all the '**.mp4**' files in the '**/ home' directory that** are more than 10MB and delete them with a single command, run:

```
find /home -type f -name *.mp4 -size +10M -exec rm {} ;
```