# Some Basic Linux Terminal Commands…

There are many helpful lists of Linux Commands on the web but this short list is a start on your command line adventure. While many terminal commands have GUI "shortcuts" you will find these commands often get the work done faster with less effort!

One of my favorite short cuts is the up-arrow and the down-arrow. Every time you type a command in Linux (or Unix) it is saved in a hidden file in your directory. The next time you need to use it again just press the up-arrow to recall the command and press return to execute. If you pressed the up-arrow too often and want to go back to one that has already been on the screen just press the down-arrow. BTW the list saved is finite. Enjoy!

## #1: apt-get

**Useful For: Managing packages**

Apt — Advanced Packaging Tool — is the single most important command on this list because it's the one you use to manage packages. It doesn't matter if your run a GUI or not: if you use Ubuntu, you use apt.

Apt-get has been replaced by the simpler 'apt' in Ubuntu 16.04 (though both work). At the time of our poll this wasn't promoted, or indeed enabled in 15.10. Forgive its omission here.

Some example **apt** commands:

```
sudo apt install application-name
sudo apt-get remove application-name
sudo apt-get autoclean
```

See the apt man page for further information on its usage.

## #2: ls

**Useful For: Finding out which files are where**

When you want to find a file, or get a quick overview of what files exist in the current directory, you can use the ls command (ls is short for '**ls**t').

Using **ls** on its own, with no flag, will list the names of files and folders within the current directory. This omits information like name file, format, size, date modified, etc.

To see directory contents with some of its data in a human readable format use the '-lh' flag, like so:

```
ls -lh
```

# Some Basic Linux Terminal Commands…

You can sort files based on size (largest file size to smallest) by passing the '-lS' flag (that's a lowercase l and a capital S):

```
ls -lS
```

See the ls man page for more things you can do with this command.

### #3: cd

**Useful For: Moving around your filesystem**

The cd command, also known as chdir (change directory), is a command used to change and navigate through directories.

Cd will assume you're in your Home folder (unless otherwise listed).

Its use is straightforward. To 'change directory' from Home to the Pictures folder you would run:

```
cd Pictures
```

Then you could run a subsequent command in this folder – e.g, 'mkdir' to create folder, 'ls' to list files, and so on.

Now let's go into another folder within Pictures:

```
cd cats/
```

To go back to your previous directory add a hyphen suffix, like so:

```
cd -
```

To move back one directory add '..', like so:

```
cd ..
```

To go back to your root/home directory simply run:

```
cd
```

See the manpage for this command to learn more about its uses.

### #4: sudo

**Useful For: Doing ninja stuff**

*Sudo… Super Do… Super User…* Whatever you call it, you can't do anything too dramatic to your system without it. That makes it possibly the single most important command on this list.

# Some Basic Linux Terminal Commands…

**sudo** lets you run commands, install software, edit protected files, as the superuser. It requires authentication using your user or root user password.

Example commands:

```
sudo edit /usr/share/applications/application.desktop
sudo apt-get install application-name
```

The related command **sudo !!** was also suggested a number of times. This is one of my own personal favorites as it lets you (quickly) run the previous command entered as root when/if you forget to add it in.

```
apt install corebird
```

```
sudo !!
```

See the man page for sudo to learn more.

## #5: cat

**Useful For: Seeing what a file contains**

**cat** stands for "catenate"

The cat command read data from files and outputs its content in the terminal. Using cat is the simplest way to display file contents at the command line.

Example:

```
cat examplefile.txt
```

To see the same file but with number lines on display pass the -n argument:

```
cat -n examplefile.txt
```