

## Common Commands and Options

This guide lists common useful commands available on the linux terminal. For more information, most commands also have a 'manual' page that lists the various flags, inputs and outputs. There are two common ways to access the command's manual:

1. `man [command]`
2. `[command] -h`

### Inputs / Outputs:

Linux terminal commands use the "standard input" (stdin) and "standard output" (stdout), where the input is the information you pass into the command, and the output is displayed to the terminal once it is run. Operators redirect this input and output.

### Flags:

A flag is an option that can be passed into a command. These take the form of a "tick" followed by a letter or "double tick" followed by a whole word. They modify the functionality of the command

Ex: `-a` or `--all` makes the `ls` command also show hidden files

## Folder Navigation

**ls**: "List" lists all files and folders in a directory

- `-a`, `--all` lists hidden files and directories
- `-R`, `--recursive` lists subfolders and files
- `-l` lists more info about files (permissions, size, owner)

**cd** [directory]: "Change Directory" navigates user to a new directory

- `~` home directory
- `../` moves up one level
- `./` current directory
- `/` root directory
- `-` return to previous directory

**pwd**: "print working directory" prints the path to the current directory of your terminal

**nautilus** [directory]: opens the visual file manager for gnome at a given directory

- `.` opens the file manager at the current directory

## File Manipulation

- |  |   |
|--|---|
| <b>mkdir</b> [folder_name]                 | "make directory" creates a new folder in the current location |
| <b>touch</b> [file_name]                   | makes a new file in the current location                      |
| <b>mv</b> [file] [new_location]            | "move" moves a file to a new location                         |
| <b>cp</b> [file] [new_location]            | "copy" copies a file to a new location                        |
| <b>rm</b> [file_name]                      | "remove" deletes a file (WARNING: this is PERMANENT)          |
| <code>-r</code> , <code>--recursive</code> | "removes" all sub-folders and files (DANGEROUS)               |

## Miscellaneous

- |                                  |  |
|----------------------------------|--|
| <b>echo</b> [input]              | copies any given input into the output of the terminal |
| <b>which</b> [command]           | returns the path to the program location               |
| <b>cat</b> [file_1] [file_2] ... | "concatenate" outputs file contents to terminal        |
| <b>grep</b> [pattern] [file]     | scans a file or input for certain string patterns      |
| <b>diff</b> [file_1] [file_2]    | compares two files and outputs their differences       |

### Permissions and Access

**sudo** [command] “SuperUser DO” runs the next command as administrator (Root user)  
**chmod** [permissions] [file] gives a file a specific set of permissions  
  +*x* “executable” allows user to run the file as a program  
  +*r* “readable” allows user to view the file  
  +*w* “writable” allows user to edit the file  
  777 grants ALL permissions  
**chown** [owner] [file] changes the owner of a file  
  *root* changes the owner to the “root” (SuperUser)

APT: The “advanced package tool” is a program that can automatically install and track packages from the internet. These packages extend the functionality of your terminal by adding new commands or installing tools

**apt** [command] “advanced package tool” manages the packages on your machine, often used with **sudo** to install packages with admin privileges  
  *update* synchronize local package index with most up to date version  
  *upgrade* downloads and installs any new updates found with **update**  
  *remove* removes a package from your computer

### Bash Script Commands

These commands can be used in bash to create temporary variables and shortcuts, but are erased upon closing the terminal.

*.bashrc* is a bash script located in your home directory that is run each time you open a new terminal. This is useful for setting up common variables and aliases for future use.

**alias** [alias\_name]=[command line]’ creates a custom command that calls other command(s)  
**export** [variable\_name] creates a new environment variable, denoted with a \$

Note: these are commands for the ‘Bash’ shell, the default shell for Ubuntu installations. Other shells have different commands and options

### Python

**python** starts command-line python with your default python version  
**python3** starts command-line python with python version 3

[*file*] specifying a file runs that file as a python script (should have .py extension)

### Operators

| “pipe” transfers the output of one program into another.  
> “write” writes output of a program into a file (this OVERWRITES any existing file)  
>> “append” adds the output of a program onto the end of a file  
& runs the prior command in the background