Linux command cheat sheet

Below is a list of commonly used commands for basic linux and in particular Bash (the name of the shell /language we are using).

You will need to replace the keywords in <>. For example replace <directory> with the name of a directory.

## Common terms and characters:

**filename** **:** The name of a file

**/ :** Is the linux path separator which separates directories and files (on Windows '\' is used instead)

**path** **:** the directory structure to the file, e.g. /this/is/the/path/filename

## Common file extensions:

These are commonly used to denote the file contents – note that they are not required.

**.sh** : a shell (bash) script

**.py** : a python script

## Common commands:

|  |  |
| --- | --- |
| **Command** |  **Action** |
| ls | List the contents of a directory |
| cd <directory> | Change directory |
| cp <file> <newfile> | Copy a file to new file. Note that “<file>” chould include the path to the file. i.e. cp myfile.txt /home/my\_nnew\_file.txt |
| rm <file1> <file2> ... | Delete files / directories |
| pwd | Print the current working directory |
| mkdir <directory> | Make a new directory with the given name |
| chmod u+x <filename>  | To make a script executable by yourself |
|  |  |

## HINTS

The TAB key can be used to auto-complete commands / paths.

The up/down cursor keys can bring up previously used commands.

Running “cd” without a directory will take you to the home directory.

Running “cd -” will take you to the previous directory.

Adding “&” to the end of a command makes it run in the background freeing up the terminal. This is useful for when executing graphical programs such as text editors.

Ctrl-C can be used to KILL a command.

To run a script in the current directory preface it with ./ e.g. ./my\_script.py

## Editing text

To edit text files there are a number of options available under linux. A user friendly option available on most systems is gedit. To open a file for editing in gedit type the following command:

gedit <filename> &