# Linux cheatsheet

## Linux basics

**Directory/folder** A location for holding files. Keeps files organized. This is like viewing a folder in Windows or Mac file explorer.

**Current/working directory** The directory in which a user is currently working

**Files** A computer file containing information

**File names** Files are often named with an extension to show their file type (e.g. a music file might be `mozart.mp3` and a picture might be `family.jpg`). Case matters for names e.g. `L.jpg` is different than `l.jpg`.
*Avoiding spaces in names will save a lot of downstream trouble.*

**Programs/commands/executables** These are several synonyms for files which are intended to be ran or executed by the computer. These files are not data but instead are commands for the computer either in binary or text. For example, think of Microsoft Word or a computer game. An example of a command is `ls` or `mkdir`.

**Options/arguments** These are modifiers to a command. They are usually specified by `-` followed by a letter or a filename. For example, `ls -l` or `ls thesis`.

### Directory shortcuts

| | |
|---|---|
| `.` | The current "working" directory (often unnecessary to specify this). It is often unnecessary to specify this since it is assumed by default. |
| `..` | The parent directory of the current directory e.g. if you are in `/var/root` then `..` is `/var`. This is like "Up" in Windows file explorer. |
| `~` | Your home directory (different for each user). On this system, this is `/var/root`. This is like the "My Documents" directory in Windows or "Home" on Macs. |
| `-` | The last directory. Use as a shortcut to go back to the previous directory you were in. Like the back button in browsers or file explorers but limited to one back. |

### Handy shortcuts

**Tab completion** When typing, you can press the `Tab` key to have the computer automatically fill in the rest of a command or file. For example, typing:
`ls hel`
then hitting Tab will autocomplete to `ls hello.c` if there is a file called `hello.c` in the current directory.

If there are multiple files that could complete, then nothing will happen but pressing `Tab` twice will display the list of possible completions. For example, you can type:
`l`
then hit Tab twice to see a list of commands that start with `l`. You can then continue typing until you reach a point where what you have typed is unique and hit `Tab` to complete.

**History** The command line stores a history of what you have entered. You can press the `Up` and `Down` keys to scroll through your history. For example, if you make a typo:
`mkdir thessi`
you can press `Up` and correct the misspelling without retyping the whole command.

## Listing contents

### `ls`: list files

To list the contents of your current directory type:

```
ls
```

To list the contents of a directory named `thesis` type:

```
ls thesis
```

To list all files in the current directory ending in `.txt` type:

```
ls *.txt
```

There are several useful arguments to `ls` (note that lower or UPPER case matters):

| | |
|---|---|
| `-l` | Show the contents as a detailed list instead of just file names |
| `-a` | Show all files in the directory. By default files starting with `.` are not shown by `ls`. |
| `-S` | Size sort the files. Combines well with `-l`. Useful to find big files in a directory. |
| `-t` | Sort the files by the time of last modification. Combines well with `-l`. Useful to find a file you (or a program or a collaborator) recently modified in a directory. |
| `-h` | Show file sizes in human readable format (e.g. 1G instead of 1000000000 or 10K instead of 10000). Really only useful with `-l`. |

# Directories (folders)

## `pwd`: print working directory

To find out what directory you are currently working in type:

```
pwd
```

## `mkdir`: make a directory

To make a directory named `directoryName` type:

```
mkdir directoryName
```

To make a directory named `thesis` type:

```
mkdir thesis
```

## `rmdir`: remove a directory

To remove a directory named `thesis` type:

```
rmdir thesis
```

Note that the directory must be empty otherwise rmdir will not remove the directory.

# Getting help

## `man`: getting the manual

To get help on a command within the command line you can type `man commandName`. For example, to get help on `ls` you can type:

```
man ls
```

Unfortunately, this help is often designed for people who already have a pretty good idea of what they are doing and may be less helpful for a novice.

## Google

Programmers like to talk about programming and are usually internet savvy so there are often lots of helpful webpages if you can figure out the right search terms. For example, a google search of "creating a directory in linux" turns up many examples of `mkdir`. Adding "tutorial" to a search will often pull up even more detailed explanations. Adding "linux" is often necessary to filter out other operating systems and synonyms.