# Desktop Environment

1. Before DE there was CLI
    1. there are many graphical desktops in linux
        - GNOME
        - KDE
        - XFCE
        - MATE
        - BUDGIE
        - LXDE
        - Cinnamon
        - Openbox
        - LXQT
        - Pantheon
        - Deeping DE
        - Fluxbox
2. GUI: Graphical interface, like MACOS and Windows
3. DE: environment implemented made oa bundle of programs on top of the OS
4. On Windows and MACOS the user is limited to one GUI and DE, on linux you can have multiple

# The GNOME Desktop Environment

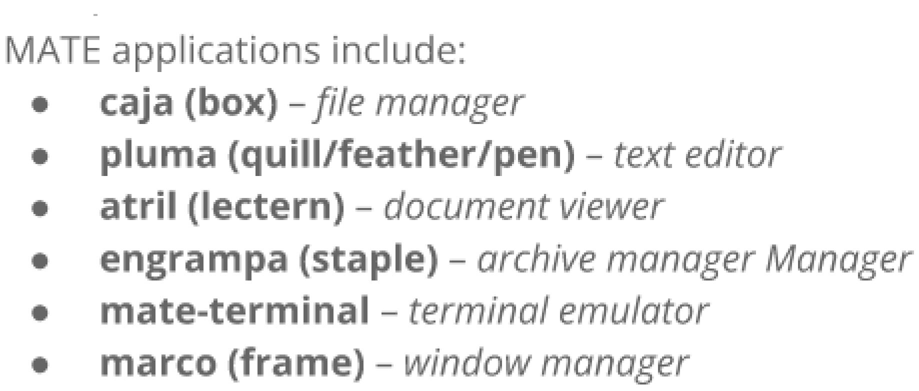
1. The default desktop in Ubuntu is GNOME 3, its also used by many more distros.
2. GNOME is short for GNU Network Object Model Environment
3. Its part of the GNU project and developed by volunteers and paid contributors
4. started as a free software in August 15, 1997, by Miguel de Icaza and Federico Mena

# The KDE Destop Environment

- The Kool Desktop Environment started in 1996 and it first released in 1998
- KDE had additional software projects and was rebranded as KDE Plasma in 2009

# Other Desktop Environments

- XFCE is a lightweight DE that aims to be fast and using low system resources while still being user friendly and good looking
- started by Oliver Fourdan in 1996
- has a linux flavor called Xubuntu
- The MATE DE is a continuation fo GNOME 2
- has forked apps from the Gnome Core Apps and others have been written from scratch



MATE applications include:

- **caja (box)** – *file manager*
- **pluma (quill/feather/pen)** – *text editor*
- **atril (lectern)** – *document viewer*
- **engrampa (staple)** – *archive manager Manager*
- **mate-terminal** – *terminal emulator*
- **marco (frame)** – *window manager*

---

- Cinnamon is a free and open-source DE for the X window System that comes from GNOME 3 but still follows traditional DE methods
- Cinnamon is the main DE of the Linux Mint distro
- its development started as a response to the release of GNOME 3 and the decision to drop GNOME 2 support
- easy to use with gentle learning curve
- the LXQT DE is a lightweight Qt DE
- included in most linux and BSD distros
- lightest yet fully functional DE
- DEEPIN DE is the DE of the chinese distro of Deepin
- built on Qt and available for multiple distros
- the devs maintain their own window manager called dde-kwin
- Pantheon DE was created for the Elementary OS
- written from scratch in BALA using GTK 3 and Granite
- Similarities with GNOME shell and MACOS
- the linux answer to MACOS

# The Bash Shell

- Allows large-scale IT possible.
- necessary component for modern computing
- shipped with almost every computer in the world thanks to Brian Fox
- CLI:
  - Command line interface meant to interact with a computer program by using lines of text
  - two ways to access CLI
    - Terminal Emulator
    - Linux Console
- a way to access the CLI is taking the linux system out of a GUI and making it in text mode
- Linux console emulates the old days of hard wired terminals
- when linux starts it creates multiple virtual consoles
- a virtual console is a terminal session that runs in the linux system memory
- most linux distros start with five or six VCs that you can access
- VCs can be accessed by using a single keystroke combination
- text mode use the whole screen
- the number of VCs are displayed with the acronym TTY plus the number

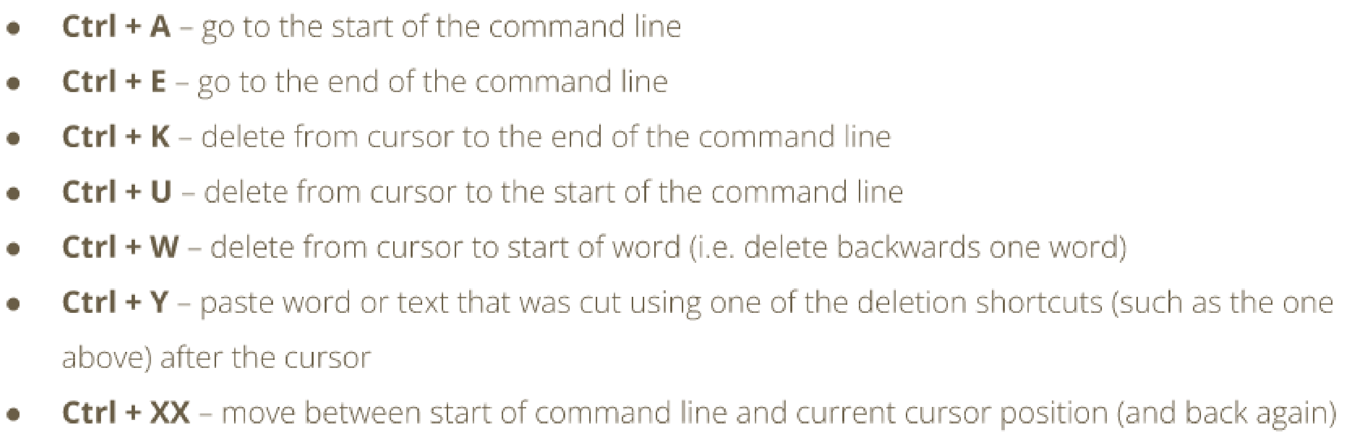
- you can log into the terminal using ID after the login and password
- within the linux VC you don't have the ability to run any graphical programs
- after you have logged in you can keep it active and be able to switch to a different VC without losing you active session

## Terminal Emulator

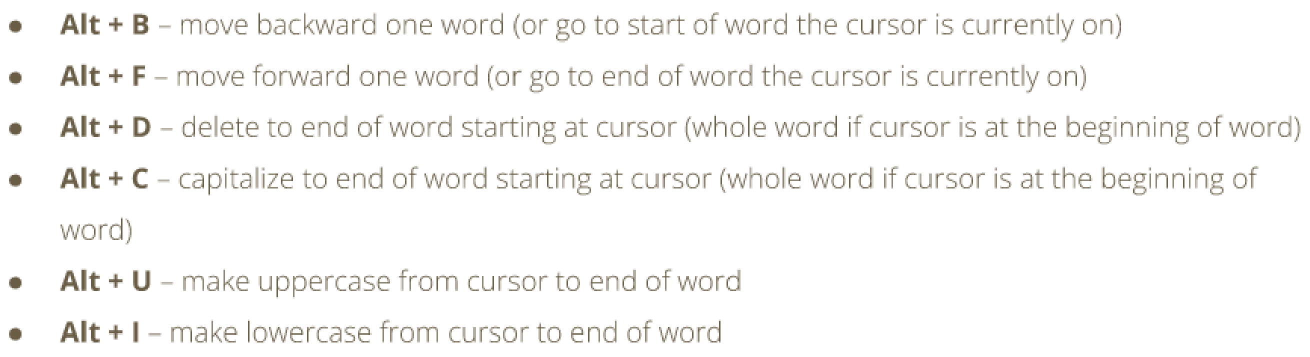
- a program that allows the access to the linux CLI. Will be used most of the time if you have a GUI
- some emulators are:
  - GNOME Terminal (included in Ubuntu 20.04)
  - Konsole
  - Terminology
  - RXVT-Unicode
  - TILIX

## The Bash shell

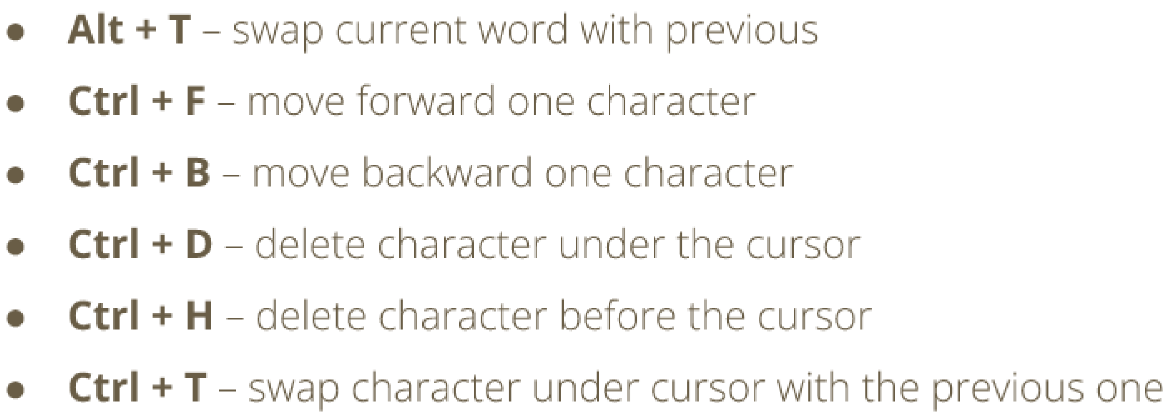
- the GNU bash shell is a program that provides interactive access to the Linux System
- It runs as a regular program and is normally started whenever a user logs into a terminal
- other shells exist:
  - TCSH shell
  - Csh Shell
  - Ksh Shell
  - Zsh Shell
  - Fish Shell



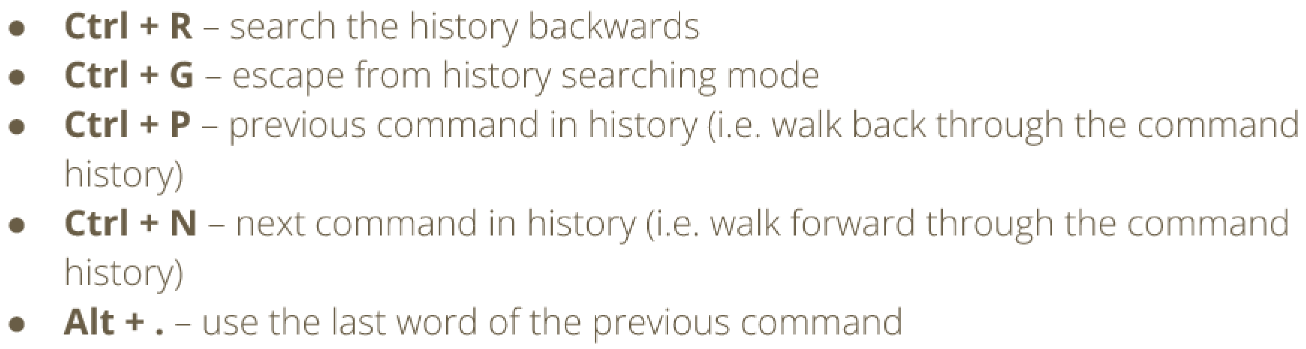
- **Ctrl + A** – go to the start of the command line
- **Ctrl + E** – go to the end of the command line
- **Ctrl + K** – delete from cursor to the end of the command line
- **Ctrl + U** – delete from cursor to the start of the command line
- **Ctrl + W** – delete from cursor to start of word (i.e. delete backwards one word)
- **Ctrl + Y** – paste word or text that was cut using one of the deletion shortcuts (such as the one above) after the cursor
- **Ctrl + XX** – move between start of command line and current cursor position (and back again)



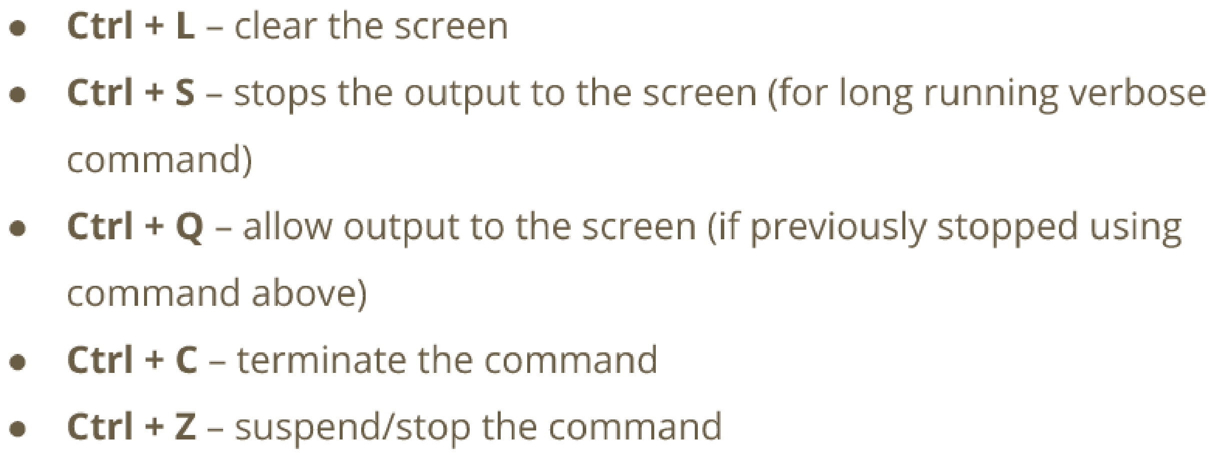
- **Alt + B** – move backward one word (or go to start of word the cursor is currently on)
- **Alt + F** – move forward one word (or go to end of word the cursor is currently on)
- **Alt + D** – delete to end of word starting at cursor (whole word if cursor is at the beginning of word)
- **Alt + C** – capitalize to end of word starting at cursor (whole word if cursor is at the beginning of word)
- **Alt + U** – make uppercase from cursor to end of word
- **Alt + l** – make lowercase from cursor to end of word



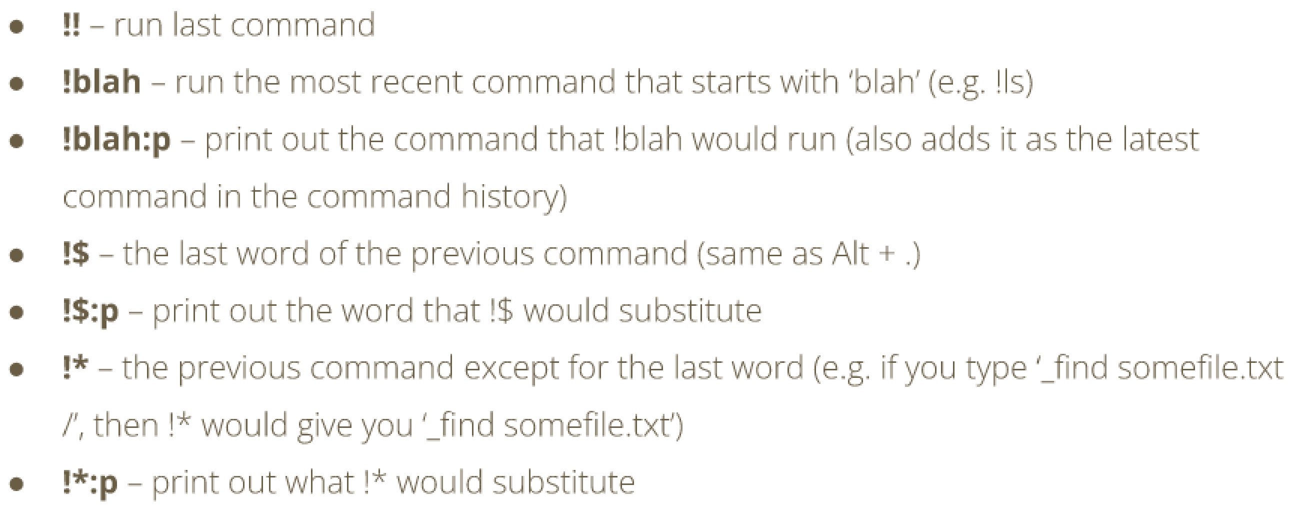
- **Alt + T** – swap current word with previous
- **Ctrl + F** – move forward one character
- **Ctrl + B** – move backward one character
- **Ctrl + D** – delete character under the cursor
- **Ctrl + H** – delete character before the cursor
- **Ctrl + T** – swap character under cursor with the previous one



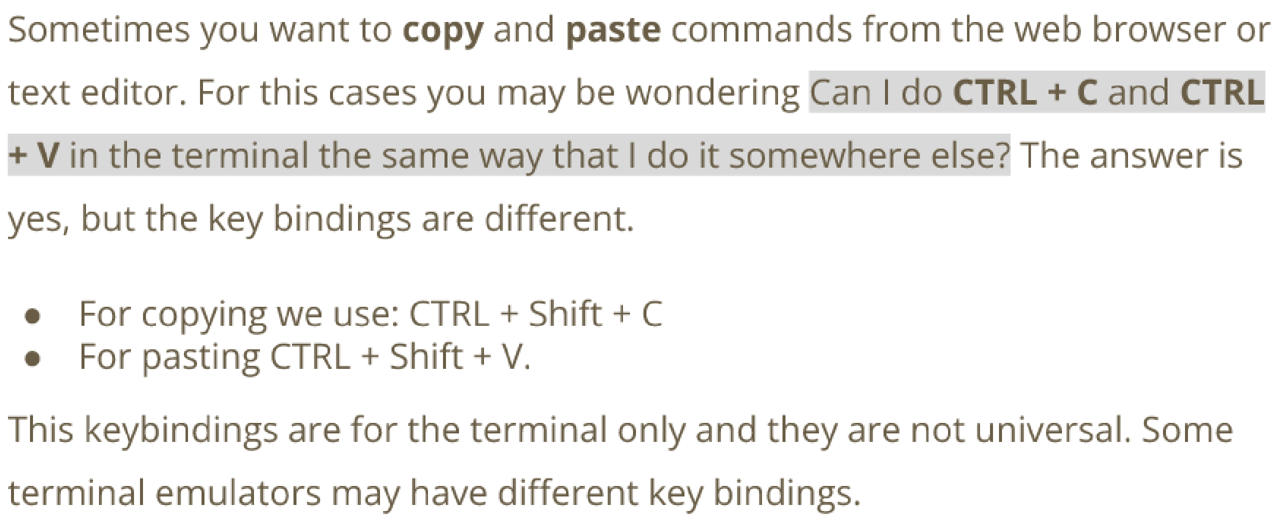
- **Ctrl + R** – search the history backwards
- **Ctrl + G** – escape from history searching mode
- **Ctrl + P** – previous command in history (i.e. walk back through the command history)
- **Ctrl + N** – next command in history (i.e. walk forward through the command history)
- **Alt + .** – use the last word of the previous command



- **Ctrl + L** – clear the screen
- **Ctrl + S** – stops the output to the screen (for long running verbose command)
- **Ctrl + Q** – allow output to the screen (if previously stopped using command above)
- **Ctrl + C** – terminate the command
- **Ctrl + Z** – suspend/stop the command



- **!!** – run last command
- **!blah** – run the most recent command that starts with ‘blah’ (e.g. !ls)
- **!blah:p** – print out the command that !blah would run (also adds it as the latest command in the command history)
- **!$** – the last word of the previous command (same as Alt + .)
- **!$:p** – print out the word that !$ would substitute
- **!*** – the previous command except for the last word (e.g. if you type ‘_find somefile.txt /’, then !* would give you ‘_find somefile.txt’)
- **!*:p** – print out what !* would substitute



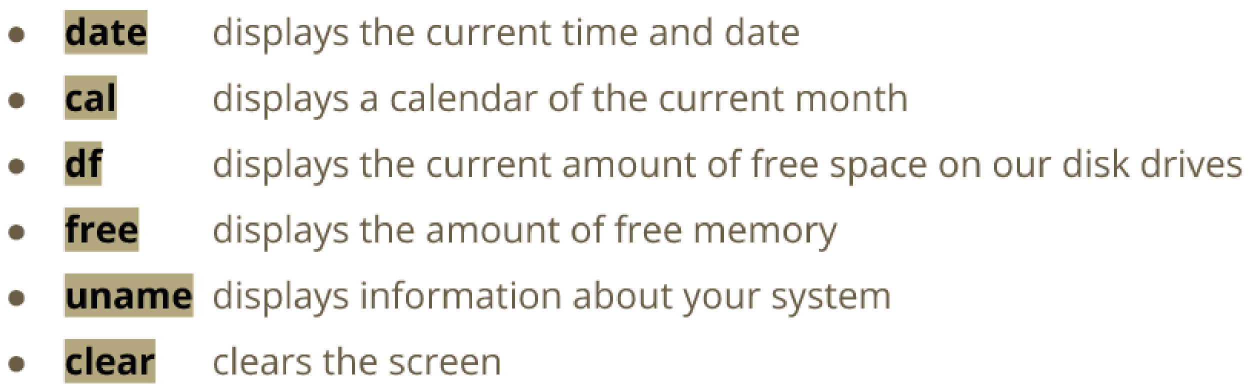
Sometimes you want to **copy** and **paste** commands from the web browser or text editor. For this cases you may be wondering Can I do **CTRL + C** and **CTRL + V** in the terminal the same way that I do it somewhere else? The answer is yes, but the key bindings are different.

- For copying we use: CTRL + Shift + C
- For pasting CTRL + Shift + V.

This keybindings are for the terminal only and they are not universal. Some terminal emulators may have different key bindings.

Shell Prompt

- the shell prompt is what appears when a terminal is launched and the shell is ready to accept input
- can varie in appearance depending on the distro
- if there is a $ instead of a # you have logged in as a superuser (sudo) privilege



| | |
|---|---|
| **date** | displays the current time and date |
| **cal** | displays a calendar of the current month |
| **df** | displays the current amount of free space on our disk drives |
| **free** | displays the amount of free memory |
| **uname** | displays information about your system |
| **clear** | clears the screen |

- the shell keeps a log of the commands that you run
- the command history can be accessed with the up and down arrow keys
- commands can be modified

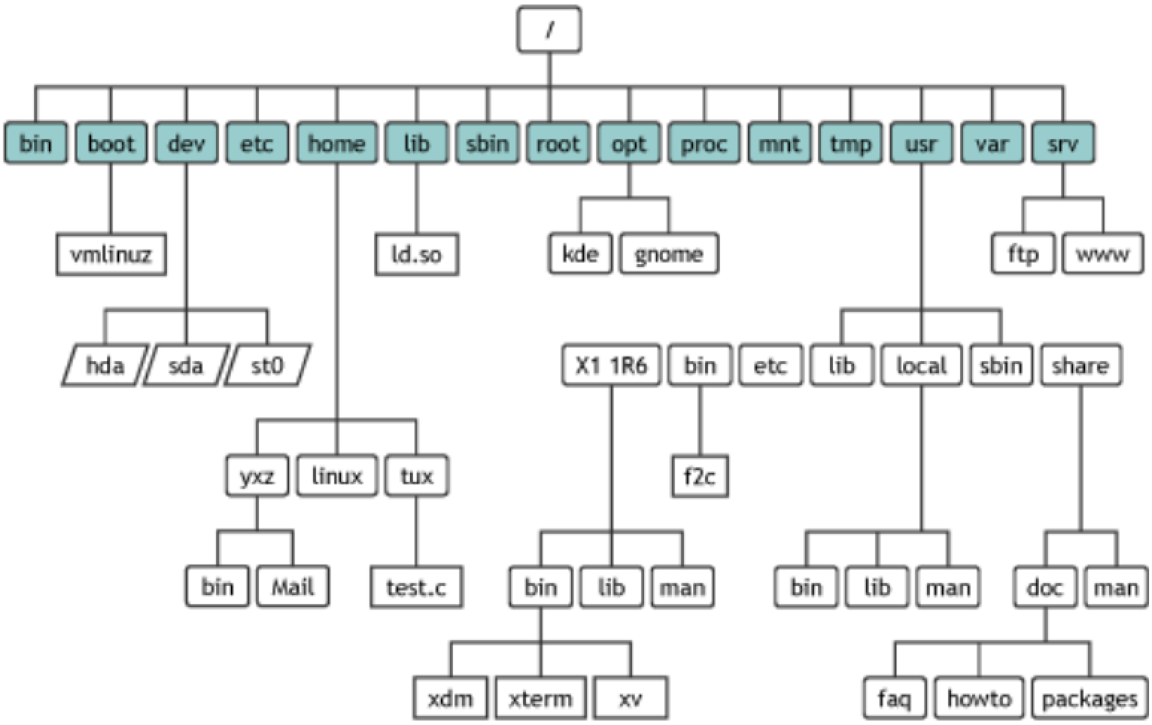
- you can run the history command to see the list of command history
- commands can also be rerun using !

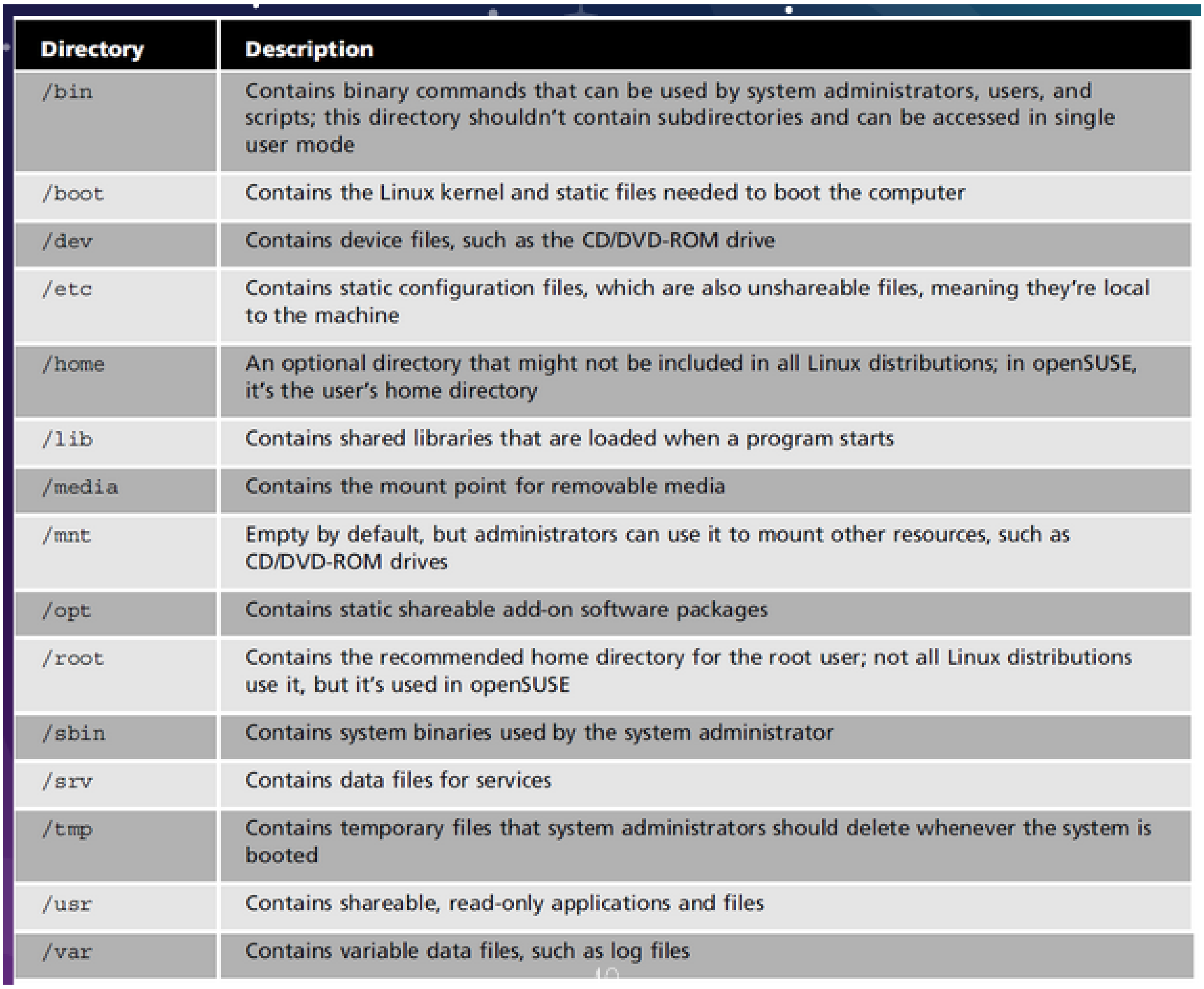
# How to navigate the filesystem

- files are stored to organize and simplify access to data
- linux organizes files in a hierarchical directory structure
- folder and directory are the same thing
- the root is the first directory in the file system
- linux always has a single file system tree
- the nemo file manager uses a GUI

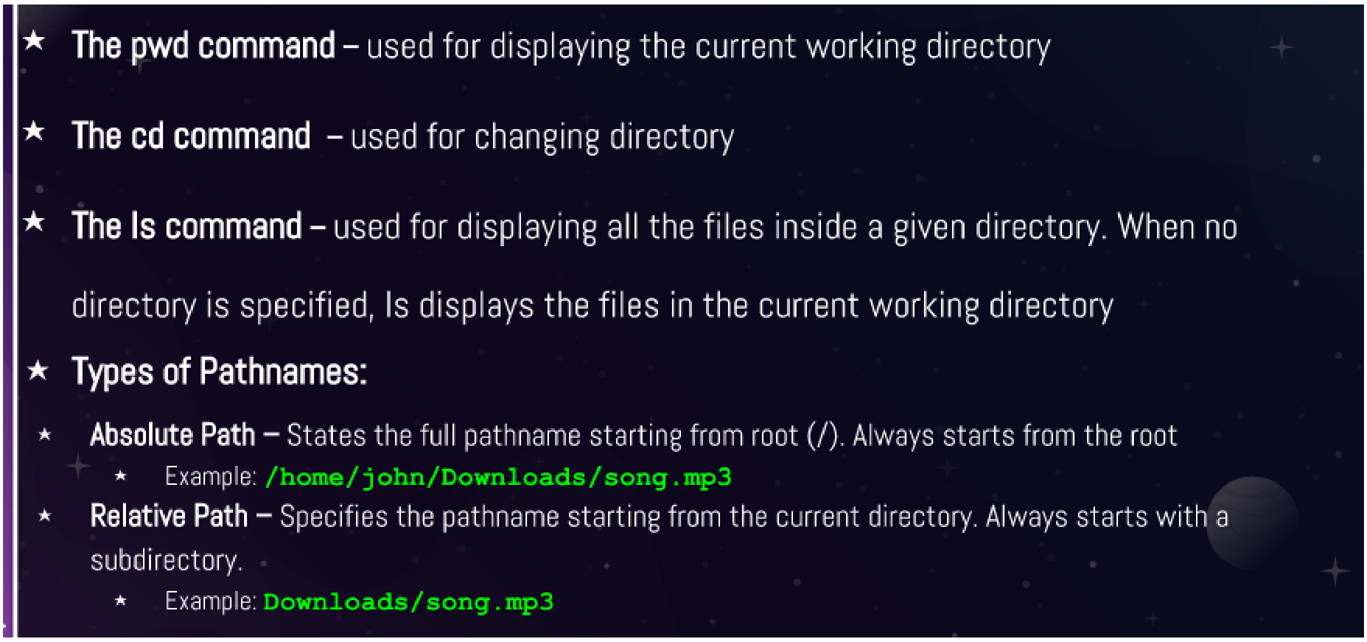
## Navigating the FS in the CLI

- the file system is like a tree where every branch is a directory
- there are parent directories and child directories or subdirectories
- every file has a pathname in the filesystem





| Directory | Description |
|---|---|
| /bin | Contains binary commands that can be used by system administrators, users, and scripts; this directory shouldn't contain subdirectories and can be accessed in single user mode |
| /boot | Contains the Linux kernel and static files needed to boot the computer |
| /dev | Contains device files, such as the CD/DVD-ROM drive |
| /etc | Contains static configuration files, which are also unshareable files, meaning they're local to the machine |
| /home | An optional directory that might not be included in all Linux distributions; in openSUSE, it's the user's home directory |
| /lib | Contains shared libraries that are loaded when a program starts |
| /media | Contains the mount point for removable media |
| /mnt | Empty by default, but administrators can use it to mount other resources, such as CD/DVD-ROM drives |
| /opt | Contains static shareable add-on software packages |
| /root | Contains the recommended home directory for the root user; not all Linux distributions use it, but it's used in openSUSE |
| /sbin | Contains system binaries used by the system administrator |
| /srv | Contains data files for services |
| /tmp | Contains temporary files that system administrators should delete whenever the system is booted |
| /usr | Contains shareable, read-only applications and files |
| /var | Contains variable data files, such as log files |



★ **The pwd command** – used for displaying the current working directory

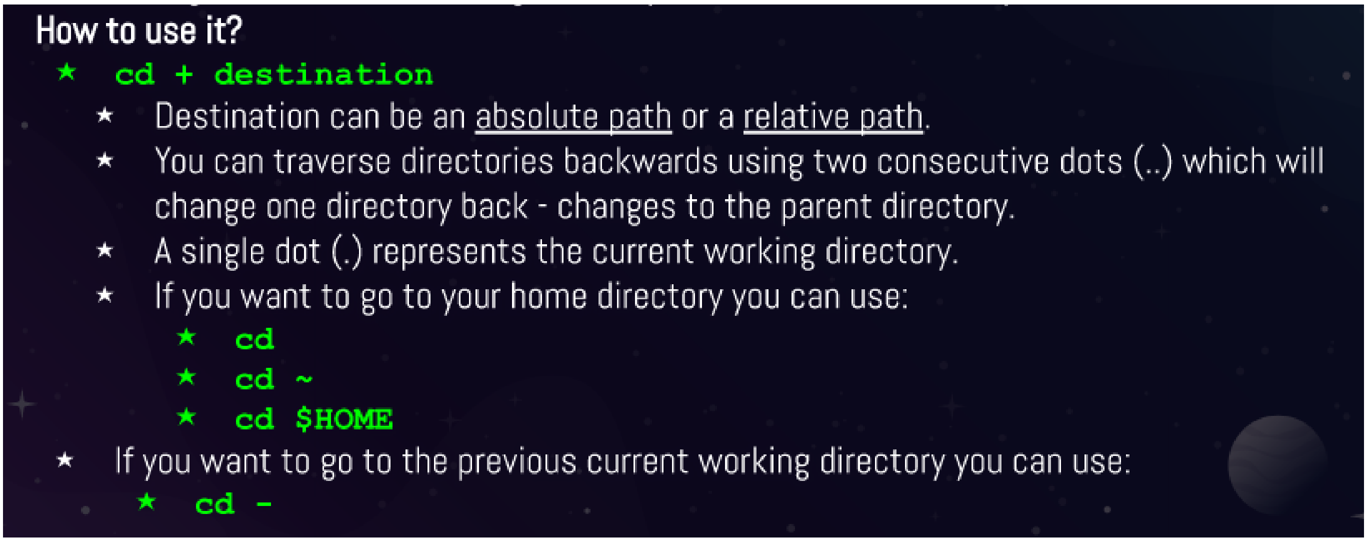
★ **The cd command** – used for changing directory

★ **The ls command** – used for displaying all the files inside a given directory. When no directory is specified, ls displays the files in the current working directory

★ **Types of Pathnames:**

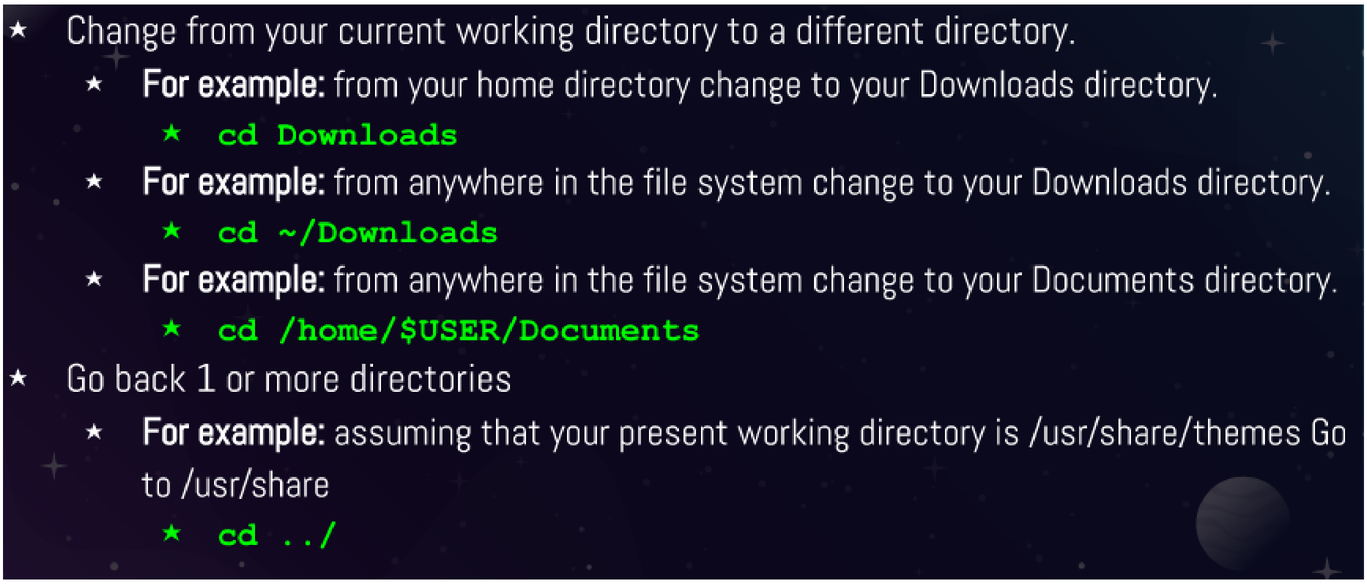
- ★ **Absolute Path** – States the full pathname starting from root (/). Always starts from the root
  - ★ Example: `/home/john/Downloads/song.mp3`
- ★ **Relative Path** – Specifies the pathname starting from the current directory. Always starts with a subdirectory.
  - ★ Example: `Downloads/song.mp3`

- PWD displays the current working directory
- CD command changes the current working directory
- it can also take you to the home directory



How to use it?

- `cd + destination`
  - Destination can be an absolute path or a relative path.
  - You can traverse directories backwards using two consecutive dots (..) which will change one directory back - changes to the parent directory.
  - A single dot (.) represents the current working directory.
  - If you want to go to your home directory you can use:
    - `cd`
    - `cd ~`
    - `cd $HOME`
- If you want to go to the previous current working directory you can use:
  - `cd -`

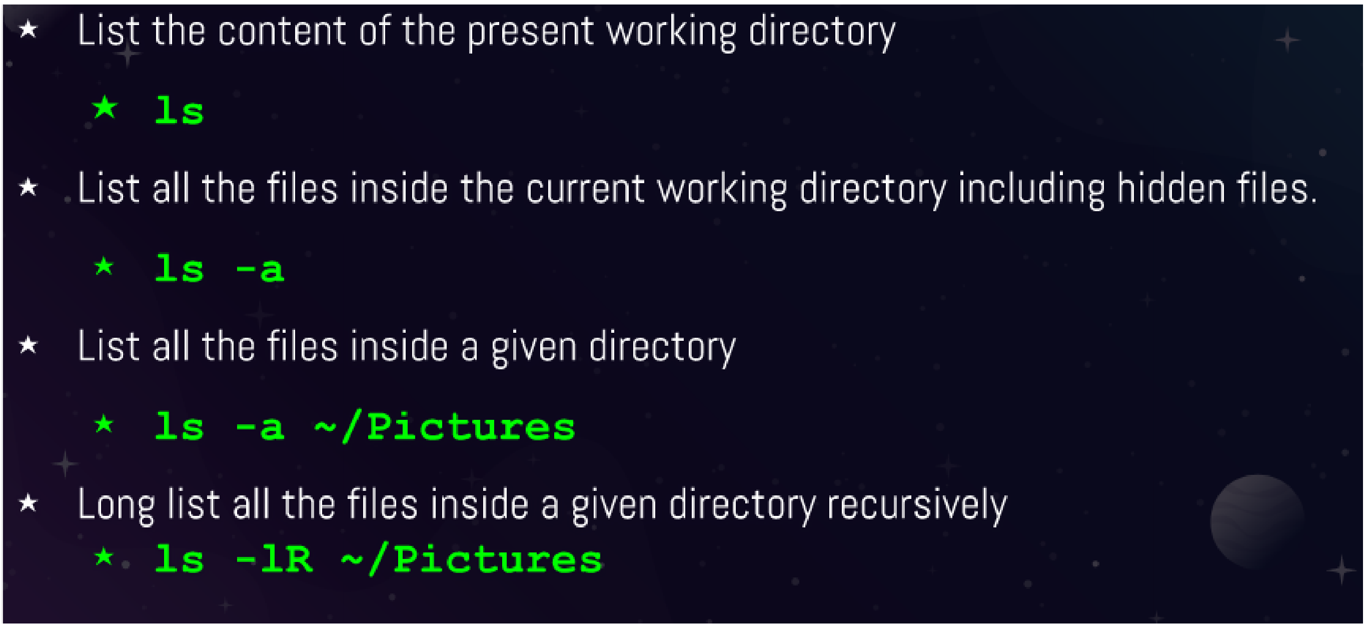


- Change from your current working directory to a different directory.
  - **For example:** from your home directory change to your Downloads directory.
    - `cd Downloads`
  - **For example:** from anywhere in the file system change to your Downloads directory.
    - `cd ~/Downloads`
  - **For example:** from anywhere in the file system change to your Documents directory.
    - `cd /home/$USER/Documents`
- Go back 1 or more directories
  - **For example:** assuming that your present working directory is /usr/share/themes Go to /usr/share
    - `cd ../`

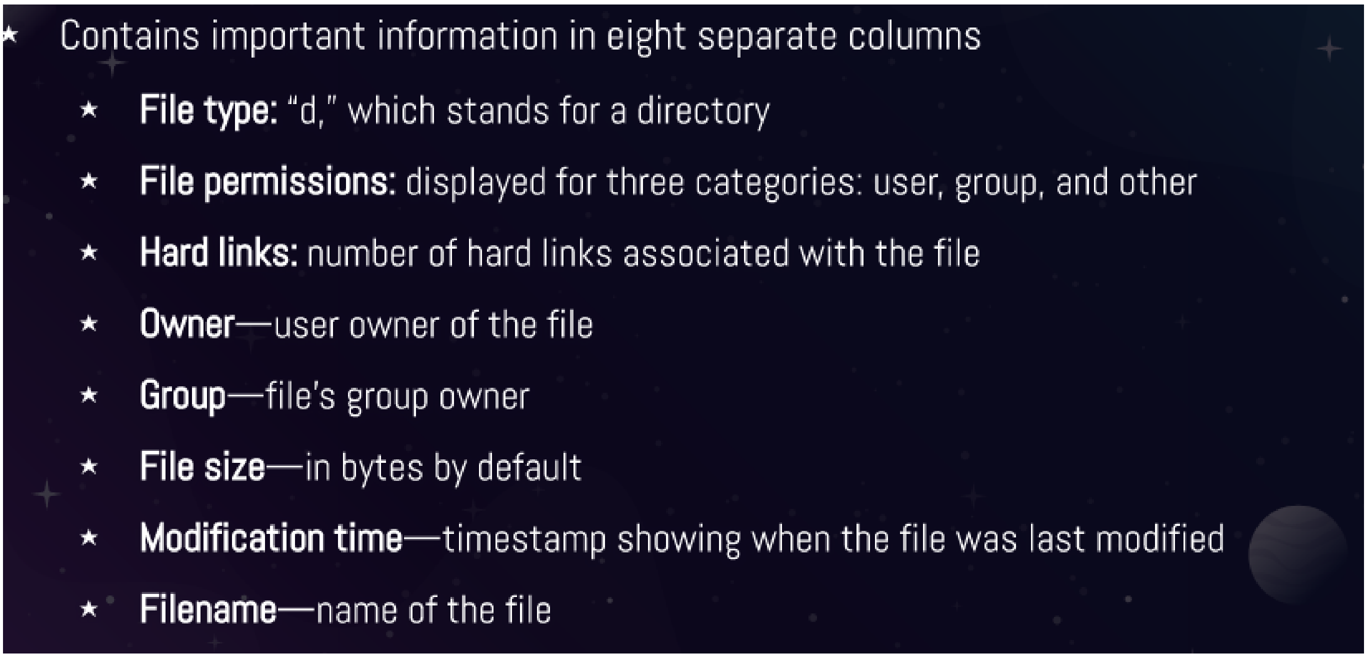
- Tab completes the command
- Arrow keys allow to move, edit and repeat commands
- **Ctrl + a** allows you to go to the start of the command line
- **Ctrl + e** allows you to go to the the end of the command line

## Listing Files and Directories

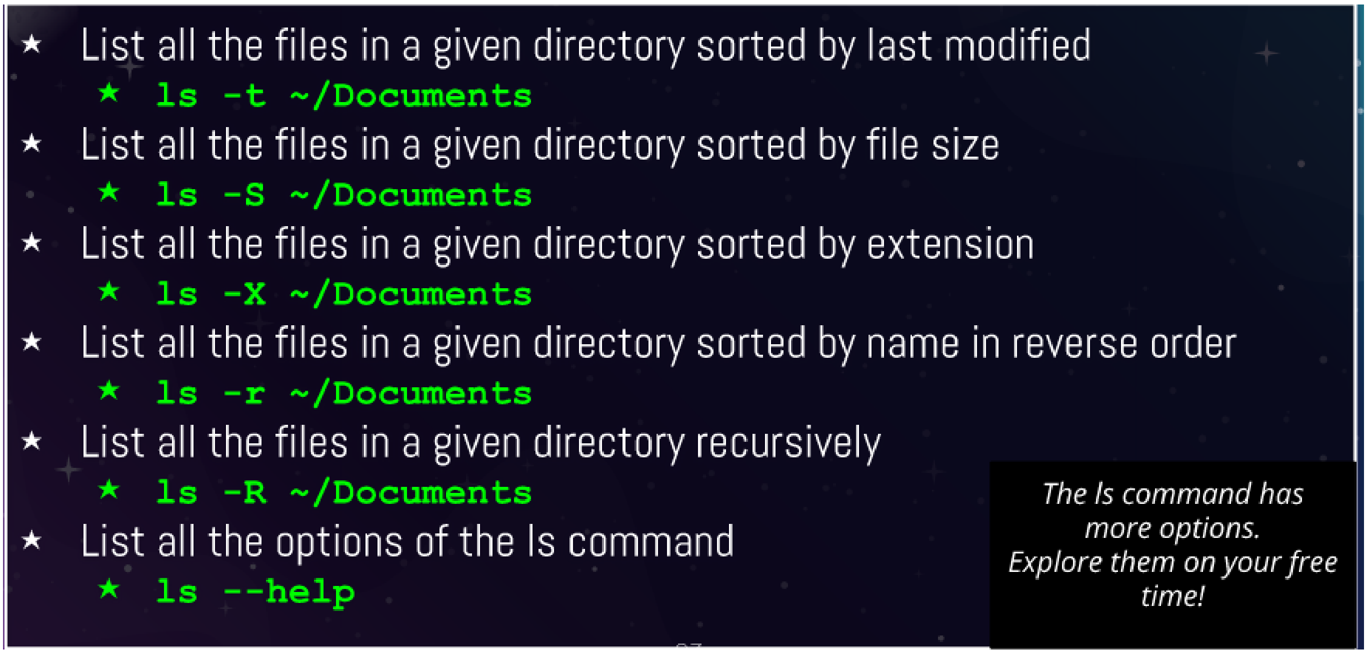
- LS command is used for listing content in the directory
- can be used with or without arguments
- has many different options and can be seen with the: man Ls command
- sorts alphabetically
- can sort in other ways



- List the content of the present working directory
  - `ls`
- List all the files inside the current working directory including hidden files.
  - `ls -a`
- List all the files inside a given directory
  - `ls -a ~/Pictures`
- Long list all the files inside a given directory recursively
  - `ls -lR ~/Pictures`

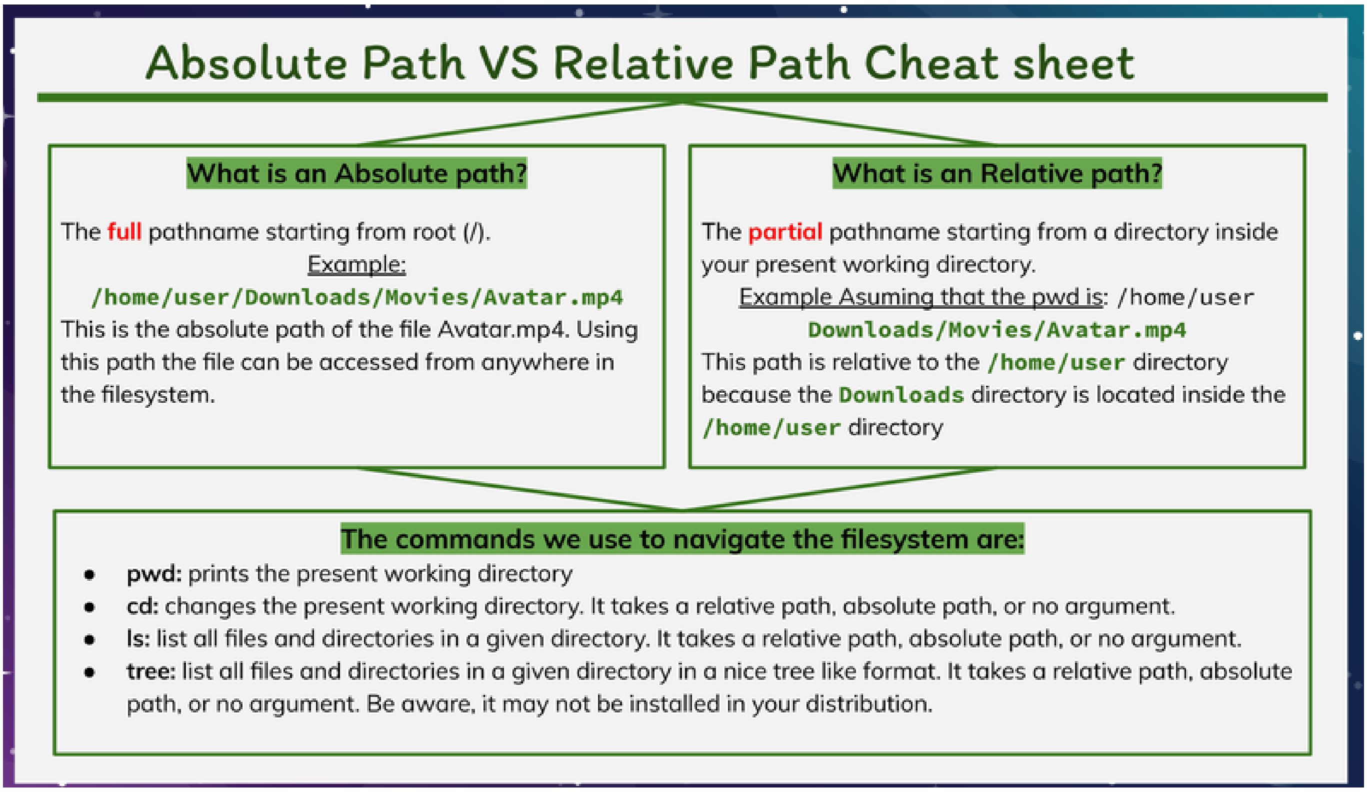


- Contains important information in eight separate columns
  - **File type:** "d," which stands for a directory
  - **File permissions:** displayed for three categories: user, group, and other
  - **Hard links:** number of hard links associated with the file
  - **Owner**—user owner of the file
  - **Group**—file's group owner
  - **File size**—in bytes by default
  - **Modification time**—timestamp showing when the file was last modified
  - **Filename**—name of the file



- List all the files in a given directory sorted by last modified
  - `ls -t ~/Documents`
- List all the files in a given directory sorted by file size
  - `ls -S ~/Documents`
- List all the files in a given directory sorted by extension
  - `ls -X ~/Documents`
- List all the files in a given directory sorted by name in reverse order
  - `ls -r ~/Documents`
- List all the files in a given directory recursively
  - `ls -R ~/Documents`
- List all the options of the ls command
  - `ls --help`

*The ls command has more options. Explore them on your free time!*



## Absolute Path VS Relative Path Cheat sheet

**What is an Absolute path?**

The full pathname starting from root (/).

Example:

`/home/user/Downloads/Movies/Avatar.mp4`

This is the absolute path of the file Avatar.mp4. Using this path the file can be accessed from anywhere in the filesystem.

**What is an Relative path?**

The partial pathname starting from a directory inside your present working directory.

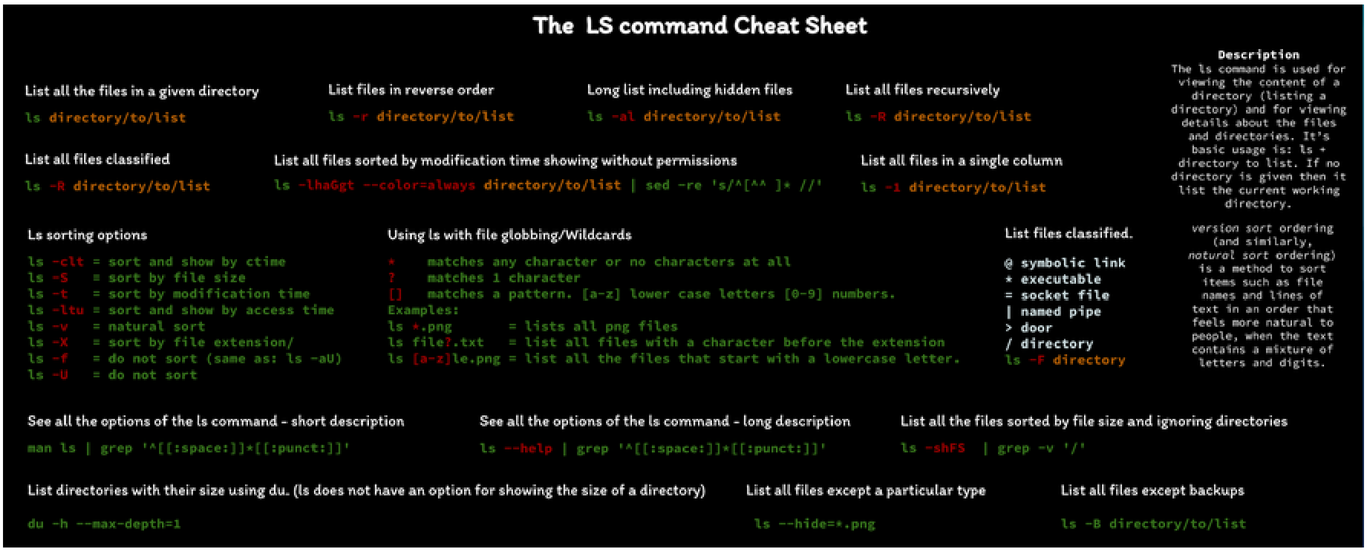
Example Asuming that the pwd is: /home/user

`Downloads/Movies/Avatar.mp4`

This path is relative to the `/home/user` directory because the `Downloads` directory is located inside the `/home/user` directory

**The commands we use to navigate the filesystem are:**

- **pwd:** prints the present working directory
- **cd:** changes the present working directory. It takes a relative path, absolute path, or no argument.
- **ls:** list all files and directories in a given directory. It takes a relative path, absolute path, or no argument.
- **tree:** list all files and directories in a given directory in a nice tree like format. It takes a relative path, absolute path, or no argument. Be aware, it may not be installed in your distribution.



## The LS command Cheat Sheet

List all the files in a given directory

```
ls directory/to/list
```

List files in reverse order

```
ls -r directory/to/list
```

Long list including hidden files

```
ls -al directory/to/list
```

List all files recursively

```
ls -R directory/to/list
```

List all files classified

```
ls -R directory/to/list
```

List all files sorted by modification time showing without permissions

```
ls -lhaGgt --color=always directory/to/list | sed -re 's/^[^^ ]* //'
```

List all files in a single column

```
ls -1 directory/to/list
```

Description
The ls command is used for viewing the content of a directory (listing a directory) and for viewing details about the files and directories. It's basic usage is: ls + directory to list. If no directory is given then it list the current working directory.

Ls sorting options

```
ls -clt = sort and show by ctime
ls -S   = sort by file size
ls -t   = sort by modification time
ls -ltu = sort and show by access time
ls -v   = natural sort
ls -X   = sort by file extension/
ls -f   = do not sort (same as: ls -aU)
ls -U   = do not sort
```

Using ls with file globbing/Wildcards

```
*    matches any character or no characters at all
?    matches 1 character
[]   matches a pattern. [a-z] lower case letters [0-9] numbers.
Examples:
ls *.png       = lists all png files
ls file?.txt   = list all files with a character before the extension
ls [a-z]le.png = list all the files that start with a lowercase letter.
```

List files classified.

```
@ symbolic link
* executable
= socket file
| named pipe
> door
/ directory
ls -F directory
```

*version sort ordering* (and similarly, *natural sort ordering*) is a method to sort items such as file names and lines of text in an order that feels more natural to people, when the text contains a mixture of letters and digits.

See all the options of the ls command - short description

```
man ls | grep '^[[:space:]]*[[:punct:]]'
```

See all the options of the ls command - long description

```
ls --help | grep '^[[:space:]]*[[:punct:]]'
```

List all the files sorted by file size and ignoring directories

```
ls -shFS  | grep -v '/'
```

List directories with their size using du. (ls does not have an option for showing the size of a directory)

```
du -h --max-depth=1
```

List all files except a particular type

```
ls --hide=*.png
```

List all files except backups

```
ls -B directory/to/list
```