Basics

- **Shell** - Text / Command line based user interface
- **pwd**
  - present working directory
  - breadcrumb trail of directories
- **ls** - list
- **cd** - change directory
- **File System**
  - APFS - Mac
  - windows - multiple root fileSystems
  - unix - one root directory, rest will be subordinate to root
    - By default in GUI some folders are hidden
- **/ - root**
- **cd /** - root directory
- **home directory (name)** - OS designates to login user
- **cd + return** - home directory
- unix is case sensitive
- **Directories**
  - **bin** - binaries, programs , we can run and accessible to all users
  - **sbin** - binaries, programs, only admin access
  - **usr** - binary executable, data non-essential to system
  - **var** - temporary files, used by various application and programs running on system
    - When systems reboots, they are not expected to remain there
    - log - who logged in computer
  - **dev** - files contain related to devices that computer can see
    - everything related to device can be accessed in form of file
  - **etc (etsy)** - system wide config files
  - **home** - other than mac, other unix systems home directory will be this
    - In mac its under Users directory
  - **Libraries** - essential third party libraries for OS and applications to work
- Almost everything in Unix which is not binary or executable, will be text file
- **ls -l** = long listing
- **ls -a** = all files including hidden files (.files)
- **"-"** switch / option (we can combine options, like ls -la)
- **dwrx** = directory, read, write, execute (permissions)
- permissions, owner_of_file, group, date_and_time
- **./** - in current directory
  - If we want to execute command which is not in well known directories (bin etc.) or PATH,
  - but still want to execute it , and are in directory where executable exists we will need (./) prepen
    to executable
- **../** - parent directory
- **cd -** = back to previous directory, even if not parent

- **cmd + t** = new window
- **cmd + w** = close current
- **cmd + q** = close all
- **touch filename** - (create file)
- **q** = quit
- **cp** - copy
  - cp filename_from filename_to
  - cp file1 file2 file3 folder (all files copied to folder in one command)
- can copy files in different directory using ../
  - same copy command, if same name no need to provide second_file name
- **-R** recursive (folder) (can be used for copying entire folder and its contents)
- **mkdir -p** (nested folders create)
  - Will create sub-folders with parent folder
- **cntrl + a** = beginning of line
- **rm** - remove ('-r' for folder) (-rf forced removal)
- **mv** - name change | (move)
  - **mv oldName newName** - name change
  - **mv file folder** - moves file between folders
- Unix has **group system** for file access, and permissions
  - read, write, execute permissions - files
  - Owner > Defined group > Other (everyone else)'
- **Group Permissions - 9 bits** (1st three - owner, 2nd three - groups, Last three - other)
  - owner | defined group | others
  - rwx | rwx | rwx
- **directory - d, - files, (@, +)** additional permissions
- **date** - modified date
- **chmod (u, g, o +, - , r, x, w)** - change mod (change file permissions)
- **chmod a=r, or u=r, g=r, o=r** (setting all permissions in one go)
  - Here we reset all permissions
  - a = all
- we can also provide instruction as binary numbers like - chmod 754
  - 421 | 421 | 421
  - we can specify combination of three bits in number form
  - -R = recursive
- **File Globs** -
  - list files starting with particular name (ls name*)
    - ls fileInitialCharcter* (* = 0 or more characters)
    - Example = ls Re*
  - list files starting with particular extension (ls *extension or *.???)
    - ls *.pdf
    - ls .(txt | pdf) = pdf or txt
    - ls *.???
  - file list using [ ] ([HJ]*) list all files starting with names H or J

- ■ ls [fm]*
  - ○ uppercase / lowercase - files / directories
    - ■ ls -d [[:upper:]]*
- **ln - s** (**symbolic link**) (a shortcut of file (like in windows))
  - ○ a pointer to file is created using above command
- **man** - manual pages, (
  - ○ **man command** - to see man page of particular command
  - ○ **space_bar** = ahead by 1 page,
  - ○ **b** = back up one page,
  - ○ **return** = one line at a time forward,
  - ○ **/** name = searches name
  - ○ **/** return = keep searching with same word
- **apropos** word = tells about which man pages & sections to look for that particular word. Alternative (man -k)
  - ○ man pages are categorised into different categories. (sections)
  - ○ each man page belong to particular section
  - ○ above command tells in which section of man pages that word can be found
  - ○ we can narrow down search to particular sections (need to do some research for mac)
- **whatis command** = little synopsis's of man command
- **whereis command** = location of command
- **clear** - clears screen
- **echo**
  - ○ work with with, without quotes
  - ○ **echo "hello world" >> file.txt** (appends hello world inside, from new line at end of file.txt"
  - ○ **\n** - new line
  - ○ **\t** - tab
- **cat** - concatenate
  - ○ **cat file.txt** (prints contents of file in standard output)
  - ○ **cat -n file.txt** (prints contents of file with line number)
  - ○ **cat file1.txt file2.txt** (we can combine contents of two file in single cat command)
- **more, less** = pagination
  - ○ we can combine results of commands with 'more' , 'less' command
  - ○ **man -k print | more** (example)
  - ○ **less file.txt** (we can paginate by using 'less' command for viewing file contents)
  - ○ **number + return** - forward (we can go ahead or backward by certain number of lines)
  - ○ **number + b** - backward (we can go ahead or backward by certain number of lines)
  - ○ **/ (search term)** - to search for word in contents of file
- **date** = current date and time (maybe different in remote servers)
  - ○ **date -u** (UTC time)
- **df -h** = disk information (FileSystem storage information)
  - ○ How much space is being used
  - ○ What drives are mounted on a unix system
- **du -h** = directory space information (numbers get rounded up)

- disk utilisation of particular directory
- **du -ah** = file information
- **du -sh** - total size consumed by particular directory