

## Final Notes

### How to clone a GitHub repository.

---

- git clone github website via tilx.

### How to use the git commands?

---

- open up tilx: git pull git add . git commit -m "message" git push

### How to write a Markdown file that contains images and proper formatting:

---

- to format any text as heading in Markdown? images
  - start the line with a # symbol then a space.

### d. How to convert a Markdown file to PDF?

---

- right click on document and convert to Pdf.

### 1. How to compress (zip) a directory/folder in Debian.

---

- navigate to the folder and right click on it. compress name and create it.

### What are Absolute paths and relative paths?

---

#### Absolute Path

---

- the location of a file starting at the root of the file system.

Example: /home/tdw/Downloads/song.mp3

---

#### Relative Path

---

- the locations of a file starting from the current working directory or a directory that is located inside the current working directory.

Example: Downloads/song.mp3

---

(provide examples with commands. For example,

---

creating a file using an absolute path.)

`cd /home/$USER/Documents`

---

How to work with the manual pages (man command)?

---

- man then the command

How to parse (search) for specific words in the manual page.

---

- `man ls | grep "specific command"`

How to redirect output (>, >>, and |)

---

How and when to redirect the output of a command to another (pipes)

---

- The pipe line allows you to redirect standard output of a command to the standard input of another.

## Usage

---

- `command_1 | command_2 | command_3 | .... | command_N |`
- 

## Basic Examples

---

- Use grep to look for a string in a particular man page
- `man ls | grep "human readable"`

How to append the output of a command to a file

---

## example

---

- `ls -la > allmyfiles.lst`

## How to use echo and output redirection to create a new file that contains some text

---

## How to use wildcards (For copying and moving multiple files at the same time)

---

## Move all files from one directory to another

---

- `mv ~/downloads/Nature/* ~/Pictures/wallpapers`
- 

## Copy specific files based solely on their file extension

---

- `cp ~/Downloads/homework/.pdf ~/Documents/.txt ~/Projects/school/` For creating entire directory structures in a single command)
- is a feature of the bash shell that generates argument strings.
- Start with open brace
- with no spaces, type your string separating entries by a command
- close the brace
- 

## Example

---

- `mkdir -pv example_site/{assets/large,docs/share,scripts/js}`

## How to create a simple “hello world” shell script

---

```
echo "hello world"
```

## How to use variables in a shell script

---

```
#!/bin/bash echo "The current User: $USER" echo -e "The Path VAR: \n $PATH"
```

## awk

---

- is a scripting language used for processing and displaying text. Awk can work with a text file or from the standard output.

## Usage

---

- awk plus options plus {awk command} plus file plus file to sav (optional)

## . Basic Example

---

Print the first column of every line of a file

---

- `awk '{print $1}' ~/Documents/Csv/cars.csv`

## . Print first field of /etc/passwd file

---

- `awk -f: '{print $1}' /etc/passwd`

Print the last field of the /etc/passwd file

---

- `awk -F: '{print $NF}' /etc/passwd`

Print the first and last field of the /etc/passwd

---

- `awk F:{print $1, " = ", $NF}`
- 

Print the first and 3 field wirth line numbers

---

- `awk -F: '{print NR,$1,$3}' /etc/passwd`

## cat

---

usage: cat plus option plus fileS(s) to display

---

Basic example display the content in a file located in  
~Documents/sample\_file

---

~Documents/sample\_file/Code/helloWorld.py

Display the content of the file with line numbers

---

`cat -n ~/Documents/sample_files/Code/helloWorld.py`

Display the content of a file including non printing characters and line endings

```
cat -A ~/Documents/sample_file/Code/helloWorld.py
```

## cp

---

## cut -

---

- The cut command is used to extract a specific section of each line of a file and display it to the screen

## Usage

---

cut plus option plus file(s)

## Basic Example

---

## Display a list of all the users on your system

---

- `cut -d ':' -f1 /etc/passwd`

## grep

---

- is used to search text given file. Grep works line by line basis( it matches the search criteria in an line by line basis)
- 

## Usage

---

grep plus search criteria plus file  
Search any line that contains the word "dracula" in the given file:

---

## Basic Example:

---

- `grep 'dracula' ~/Documents/dracula.txt`

Search any line that contains the word 'dracula' regardless of the case

---

- `grep -i 'dracula' ~/Documents/Books/draucula.txt`

## Display how many lines contain the matched string

---

- `grep -c 'dracula' ~/Documents/Books/dracula.txt`

## Bais Example:

---

### head

---

## Usage head plus option plus file(s)

---

## Basic Example:

---

Display the first 10 lines of a file

- `head ~/Documents/Book/dracula.txt`

## • Dispokay the first 5 lines of a file

---

- `head -5 ~/Documents/Book/dracula.txt`
- 

## Display the first 5 lines of multiple files

---

- `head -n 5 ~Txt/{dracula,war-and-peace}.txt`
- 

## Display the first line of multiple files using wildcards

---

- `head -n 1 Csv/.csv Code/.py`

## ls

---

## lsit all the given files in a directory

---

- `ls Downloads/*`
-

## List all the text files in a given directory

---

- `ls Downloads/*.txt`

List all the text files in a given directory that start with letter f

---

- `ls Downloads/f*.txt`

List all files that contain the word file in the name

---

- `ls file`

List all hidden files in current working directory

---

- `ls ./.*?`

List all the hidden files in the parent directory

---

`ls ../.*?`

List all the files that have 2 characters in the file name between letters b and k

---

- `ls B??k*`

List all the files that have a single character between letters f and l

---

- `ls f?l*`

man -

---

mkdir -

---

used for creating a single directory or multiple directories

mkdir plus the name of the directory

---

## create a directory in the present working directory

---

- `mkdir wallpapers`

## Creaaate directory in a different directory using relative path

---

- `mkdir wallpapers/ocean`
  - Create a directory in a different directory using absolute path
- 

- `mkdir ~/wallpapers/forest`

## mv

---

- moves and renames directories

## mv plus source plus destination

---

## mv plus file/directory to rename plus new name

---

## To move a file from a directory to another using relative path

---

- `mv Downloads/homework.pdf Documents/`

## To move a directory from one directory to another using absolute path

---

- `sudo mv ~/Downloads/theme /usr/share/themes`

## tac

---

## Usage tac plus option file(s) to display

---

## Basic Example:

---

## Display the content of a file located in ~Documents/sample\_files in reverse order

---

~Documents/sample\_file/Code/helloWorld.py

- `tac ~Documents/sample_file/Code/helloWorld.py`

## Display the content of multiple files in reverse order

---

- `tac ~Documents/sample_file/Code/helloWorld.py`
- `~Documents/sample_file/Code/helloWorld.sh`

## tail

---

### Usage tail plus option Plus file (s)

---

### Basic Example

---

### Display the last 10 lines of a file

---

- `tail ~/Documents/sample_files/`
- 

### Display the last 5 lines of a file

---

- `tail -5 ~/Documents/sample_files/`

### . Display the first 5 lines of multiple files

---

- `tail -n 5 Txt/{dracula,war-and-peace}.txt`
- 

### Display the first line of multiple files using wildcards

---

- `tail -n 1 CSV/.csv Code/.py`

# touch

---

- is used for creating files

## Examples

---

- To create a file called list
- `touch lsit`

### To create several files:

---

`touch list_of_cars.txt script.py names.csv`

### To create a file using absolute path:

---

- `~/Downloads/games.txt`

### To create a file using relative path (assuming you `pwd` is you home directory)

---

- `touch Downlowads/games2.txt`

# tr

---

- The `tr` command is used for translating or deleting characters from standard output

## Usage

---

- Standard output | `tr` plus option plus set plus set

## Basic Example

---

- Translate one character to another (For example a period with a comma)
- `cat file.txt | tr '.' ','`

## Translate white space into tabs

---

- `cat program.py | tr "[:sapce:]" '\t'`
-

# Translate white space into tabs

---

- `cat file.py | tr "[:space:]" ' '`

## tree

---