Terminal emulator - a program that will let us use the terminal in a graphical way
In windows - putty , linux - terminal. Windows also has a terminal.

Shell - in linux shell is basically a cmd line interface that will take my commands as input.
Example - git is a command line tool , we can see network requests , create forlder. Shell interprets our commands and tells the computer what to do.
Eampels of shells - gitbash, bombshell, zsh , bash.

Part of the terminal is command prompt where we write all of our commands.

Commands

Some path conventions (relative path)
. (dot) -> represents the current directory
..(double dot) -> represents the parent directory.

1. **ls** - shows the files or folder contained in the current folder.
   Ls -a -> shows hidden files. Every file that starts with . is a hidden file.
   ls-al-> hidden files with long details.
   Ls -l -> long details for all the files except hidden.
   Ls -r -> list of all the files that are i the subdirectories as well.
2. **Mkdir <folder_name>** - A folder is created here when we enter the name of the folder. Using this command.

Adding a new directory in between two directories

Now what we will do is suppose we are having our directories as random. Inside random we have hello directory, now i wish to create a new directory between these for that i will use the command:
mkdir -p random/<new_directory>/hello -p indicates here parent

3. **touch** - touch helps us create file. Syntax - touch <file_name>
4. **Cd** - change directory, we can change to any of the folder or file using this command.
   If I am in a folder which is suppose empty and i want to navigate into the pictures folder int that i would have to enter cd Pictures but as the file is not present then it would give me an error “No such file or directory”.
   Now i want to navigate to the parent folder then i will enter command “cd ..” to navigate to the parent folder.
5. **Cat** - short for concatinate. It is used to list all the contents in the file in the standard output.
   Syntax - cat > file_name (if this file is not present then it will be created)
   Now all the data i type goes into the file name which i entered here.
   Note - To exit this we can just use control button and then press c simultaneously.

Printing data from two files:

Syntax - cat <file_1> <file_2>
/* the contents of both the file are printed here in the sequential manner*/

Adding data from two files into the third one:

Syntax - cat <file_1> <file_2> > <final_file>
cat <final_file>
/* the contents of both the file are printed here*/

6. **Pwd** - present working directory , shows the current directory i am in.
7. **Echo -** prints the data we want it to. Suppose i want to print the string “Hello” inside a file named “file.txt”. I will use the command echo “Hello” > file.txt and we can observe that this string is entered inside it.
8. **Man** - syntax man <command_nam>
   The command which is entered, it’s functionality is displayed when used with the man command. Example - man echo.
9. Cp - to copy the contents of one file into the another file. Syntax - cp file.txt copy_file.txt
10. Mv - to move one file into another directory. syntax=> mv <file_name> <folder>. Note => we can also rename the files as we want. Suppose if we enter a file name in second argument then we can rename the first file into the second file.
11. Rm => to remove a file permanently we use the command rm. syntax=> rm <file_name>
12.

Less often used commands

1. Tr - translates the characters.
   Now in this example we are going to translate into uppercase from lowercase and enter the content into a new file name upper.txt.
   Example => - cat file.txt | tr a-z A-Z > upper.txt.
   Note - this is known as tunneling. Output of the first command is going to act as input for the second command. Also we have used redirection here the particular part of the command “ >upper.txt” refers to the redirection like the data has been redirected to this file.

Suppose we are writing long long commands again and again then we can create aliases for them. Aliases are shortcut command for certain commands that can be created.

Environment variables
Path environment variable.