When we type something in the terminal in Linux
it will try to search that command
and show it if its not found

https://www.freecodecamp.org/news/the-linux-commands-handbook/

help can be used before the command

1. whoami
2. man
3. clear or Ctrl+L
4. pwd
5. ls
6. cd
7. mkdir
8. touch
9. rmdir
10. rm
11. open
12. mv
13. cp
14. head
15. tail
16. date
17. >
18. >>
19. cat
20. less
21. echo
22. wc

32. |
33. sort
34. uniq

35. Expansions
36. diff
37. find
38. grep
39. du
40. df

2.
man -manual
man command_name gives a desc of that command
can have lots of pages
we can use space to go through them

in synopsis if something is there in [] its optional

3.
clear simply cant scroll back
but clear -x will clear but can scroll back

4. ls file/folder_name
then give the contents inside the foldername
if its a folder name it is appropriate to add a / at the end
its a file name no need of that
we can also give the full name of the path after ls

-l
use long listing format
give so much details if we use that

-a
gives all contents
starting with .

we can combine them also

6.cd
~ represents that the /home/username
cd ~ takes us to our home page
Eg : cd ~/Downloads

7.mkdir
make directory
mkdir
no need to move inside that folder
we can just use the path name both relative and absolute
it can be done only if the path is existing
else we want to create a nested root directory then we have to use -p

eg : mkdir -p winter/seeds/lettuce
no need of the existence of the parent directory

it will automatically create it- both the seeds and lettuce

8. touch is for creating the file
touch squash
similar to mkdir
can create multiple files as well
then we can separate it with spaces

Contents wont change again only the timestamp

if we create one more file with the same name, then it wont create a duplicate
but will update the timestamp of the file

9.rmdir
we can use this if the dir is empty

10.rm
can use to remove the files
multiple files deletion by separating with spaces
no recycle bin

-v gives the details of what it have done
verbose

-r
if we want to delete a non empty folder use the -r
it means recursively

-ri
make it interactive that means it will ask us permission
y for yes
n for no

11. open
open in an external text editor
mac specific

xdg for ubuntu
eg: xdg-open .

12.mv
rename
mv old_name new_name
mv can also used to move the file from one folder to another
for that the second thing should be the folder name ending with /

can move multiple files as well
rm pp kk ll Stuff/

here, pp kk ll will be moved inside the Stuff

13.cp

cp Jounal.txt newJournal.txt
the contents inside will also get copied
cant normally use cp STUFF use -r (recursively)

14. head
gives the first 10 lines of the file

head new_journal.txt -n 20
-n can used to specify the number of lines

15. tail
last 10 lines of the file
-f can be used to get real time logging

16. date
gives the day date time

17. >
redirection
it will redirect the contents to some other file

eg: data > date.txt
if we done this 2 times , it wont concatenate
it will update the previous content

eg : pwd > current_dir.txt

18. >>
it will concatenate instead of over writing
same as >

no need of fie existence
it will create new one

ls -l > contents.txt
it will give the terminal output to the file

-n to give the number of lines we cant to move

19.cat
cat actually means concatenation
but it will read all the lines as well

also concatenate 2 or more files
cat date.txt new_journal.txt

concatenating and redirecting
cat date.txt new_journal.txt > new.txt

20. less
it can be viewed and read in a good proper way
eg: less filename
have to google the synopsis

21. echo
prints in the terminal
use is
echo "sudygfsaudkyfgsaudygfvuh" > tooo.txt
new file and contents can be created

22. wc
word count
wc too.txt
      1   1   31   too.txt

explanation
1   Number of lines
1   Number of words
31   Number of bytes (characters)
too.txt   The filename

-l for only lines
-m characters
-w number of words

32. |
pipping
piping means taking the output of a command as the input of the next command
like for e.g. the output of the ls -v can be given to input of wc -l

E.g.   ls -l | wc
      date | wc
      cat 1.py 2.py | wc -l
      cat 1.py 2.py | wc -l > num.txt

33. sort
simply sorting the output in alphabetical order by default
not the actual file contents
case sensitive - caps initially

if there are a list of numbers then they are not sorted based on the largest or smallest
it check the letter order and write the first one

sort -n sorts based on the numbers
sort -un nums
sort -n numbers

sort -n numbers | wc -l

34. uniq

report or omit repeated lines

actually if there is a list of options that are repeating, then they are removed in such a fashion that they are repeating adjacently , not as a whole

uniq mainly uses as the conjunction with the sort command
because when we sort it repetitions will come adjacent and removed easily using the piping of uniq

uniq -d
only duplicated items

uniq -u
only one person liked

sort fav.txt | uniq -c
also gives the count

sort fav.txt | uniq -c | sort -nr

35. EXPANSIONS

~ /home/ubuntu
$PATH its a location that is the bin inside many folders
$USER it returns the name of the current user
* every path in that directory
*.txt means gives everything that ends with txt. E.g. ls -l *.txt
*.?? means every character that has a . and then exactly 2 chars after .
{} they create combinations that are possible
.. numbers in between

we can use that along with rm and all
rm *.??

{a,b,c}.txt
touch app.{js,html,cs,py}
create 4 files with extension and same name
ls app.*

echo {1..99}
prints all the num from 1 to 99

```
echo Day{1..99}
Day1 Day2 .... Day99
this idea can be used to create multiple directories, files, etc.



36. diff
compare 2 files
diff 1.txt 2.txt
and the output is like
24a25
at 24th line added new line 25
>mint
mint is the word added in the second file

if thirich , then
25d24
<mint

 if there are more changes all of them get listed out


37. find

. means searching dir and here it is everywhere

find .
then gives all the files its sub files etc. nested somewhere in the
folder

find . -name '7'
searching for some files that have name 7


find . -name '*7*'
get everything that has a 7 in anywhere in the name


find . -name '*.py'
get evry thing that has .py extension


find STUFF -name '*.js'
gives everything that has this extension and in the folder STUFF


we can search based on the type
type -d means directory
get only get all the dir

type -f means files
gives all the files



if use -in then case insensitive

-or, -not can be used
```

file sizes also

**********NOT COMPLETED**********
38. grep
global regular expression print
to search in files, or combine it with pipes to filter the output of another command

grep green song.txt
search for the word green in the file and display it with a color change

-n gives the line number as well

-nC 2
gives 2 lines after and before the word we finds a match

grep -r "CHICKEN" .
check every folder recursively and search for the word CHICKEN. gives the names of files as well

STUDY REGULAR EXPRESSION FINDING

39. du
disk usage
gives all the files and its sizes

-m for megabytes
-g for gigabytes

du -h | sort -h
human readable sort and gets the files in order

pipe with tail to get largest 10 ones alone

-r reverse

40. df
used to get disc usage information
print the informaytoion about the volumes mounted

df -h
df -h Desktop

41. history
gives the history of the commands we have used in the terminal

we can type !NUMBER_OF_THE_COMMAND
THEN we can run the exact same command

history | grep "cookie"

42. ps
process status
ps ax gives all the processes in the system
ps axww

processes are listed and there actually we can kill the unwanted processes
here, ps axww | greg "Visual Studio"

43. top
display and update sorted info about processes
top most cpu intensive properties

44. kill
kill -l
list of properties we can use
default kill sends term signal(terminate) so that it will terminate

SIGTERM(15) AND SIGKILL(9)
9 IS THE brutal way to kill

45. killall
can give the name of the program and kill it

46. job, bg , fg
job
background
foreground

Ctrl + c -----> Stopped
Ctrl +

study once more

47. gzip
lz77
data compression
compression leads to the data reduction of original file and we are left with a compressed file only
so if we want to store the original one
 then copy it or
gzip -k filename -----> for compressing
gzip -d filename_compressed ----------> for decompression

can compress multiple by spacing and as different

48. unzip

49. tar
combined into a single file and later we can compressed or something else
eg: tar -cf archieve.tar file1 file2
-x for unarchive
-tf for viewing the contents in that tar file

gzip -k archieve.tar

combined and compressed together
tar -czf bundle.tar lots.txt 222.py sedfsf.txt
rm -f lots.txt 222.py sedfsf.txt

tar -xf bundle.tar

49. nano
beginner friendly editor
nano filen_name
can make changes
at bottom
ctrl x
so exit
ask permissions there save it or CTR + S

ctrl W for search
enter press and go

ctrl + g is whole menu gving shortcut
crtl + k cut
ctrl + u paste

50. sleep
sleep 10
sleep for 10 seconds

51. alias
we use ls -al most of the time than using simply ls
so we can make ls -al when typing ls
that's why we are using alias

note that it will last only in the present terminal
not on a new one

alias myls='ls-la'
note that no spaces in between

to make it permanent

in ~ directory
ls -a and find .basharc
and open it

in that there is a section some more ls aliases
here we can write our own aliases

now we can open a new terminal and write the new command
or write
source .bashrc
then run the command newly created

normally single quotes
if there is extension
then ""

52. xargs
convert input from a standard input into arguments to a command

the output a command is used as the input of another command

command1 | xargs command2

cat deadPlayers.txt | xargs rm

READ man xargs FOR MORE DETAILS

53. ln
part of Linux File Systems
used to create links
pointed to another file - link
its similar to Windows shortcut

2 types ====> hard links and soft links

HARD LINKS
rarely used
have few limitations
cant link to directories
cant link to external file systems

ln <original> <link>
means create a file and write something
then cat both same output
its not a copy
its actually a pointer
changes in both makes the other change
if removed one , other will stay

SOFT LINKS
ln -s <original> <symlink>
if the original is changed, then the softink will also change and vice versa

if removed original, then the color of soft link changes
and shows that it is pointing to a directory that doesn't even existing
we cant open it
there is a pointer

why this?
didn't get the example of python
we can check it later

54. who
there are multiple users in same devices
so to identify which is logged in

55. su
switch user
su <username>
enter the password
in that single terminal window only

exit to get out of the user

touch cant be done in another users home
some of the permissions are restricted

even though i am logged in as elvis

su -l elvis then take to

56. sudo
to run a command as root user
owner of some imp files

input the admin password

57. apt install
for installing something

58. passwd
to change the password
enter the old, new, retype the password

you can lock someone else password, change someone else password
we can have so many privilages in this

passwd simply changes our

58. chown

changes ownership
every file in the os has an owner
they can do anything with that file

ls -l gives the 3rd coumn which is the owner

chown <owner> <filename>
mainly sudo chown

-R to recursively change the ownership

4th column is the group owner
membership
google that

59. groups
gives the list of groups we are the member

sudo chawn <owner>:<group> <file>

60. file permissions
-     regular file
d   directory
then 9 chars
3 set of 3 chars
owner group world
rwx   rwx   rwx

read permission
write permission
execute permission

exec means we can cd into that directory
file can be treated as a program and can be executed

61. change mod
changing the permissions
who we are changing the permissions for
what changes are we making
which permission are we setting

u - user (owner of the file)
g - group
o - others
a - all of the above

-     remove
+   add
=

```
chmod g+w file.txt
chmod a-rw file.txt




chmod octals
chmod 755 file.txt
7     5     5
111   101   101

1 for activating
0 for not activating



= means
a=r
change all the a permission to r
```