```
# CTRL + ALT + T - open terminal
# CTRL + D == exit - close the terminal
# CTRL + L - clear - clean up our space
# CTRL + left click - open link
# CTRL + C - cancel the current command

# sudo apt -y install ncal - calendar
# echo hello - repeating the word after 'echo'
# cal
  cal 2024
  cal -y - gives the cal command a '-y' option
# date - gets day of the week / d / m / time y
# cycle backwards through the commands being made - up arrow key
# history - show all past commands
# !4 (or !3, etc.) - repeats the command under this number from history
# !! - repeats previous command
# history -c; history -w  - clears the history and writes the changes

# command   =     command_name           options           inputs==operand

# echo $PATH - see the path and order in which the commands are being searched
# which cal - see in which path you can find the command 'cal'

# options have long and short versions:
  date -u (can be chained i.e.: date -u -f -g == date =ufg)
  date --universal (cannot be chained i.e.: date --universal --alabala != date --
universalalala)

# cal -A 1 12 2017 - gives the calendars for Dec 2017 and one month after - Jan
2018
  cal -A 1 -B 1 12 2017

# man -k which - searches where in the manual can be found 'which' command
# man 1 which - searches 'which' command in the 1st section
# help cd - shows documentation of how to use 'change directory'
# help + ENTER - lists all of the commands that need 'help' instead of 'man' to
show documentation
# Q - quit the man page
  which [-a]  filename ...
  []                    - means that it is optional option
  ...                         - you can enter more than one input
  <SOMETHING>           - it means that it is mandatory
  [-a | -b]             - pipe means one OR the other

# mak -k "list directory content" - we use quotes to search for all 3 words
together
# ls - list out a directory's content
  ls -lh (long list and human-readable)
# cat - concatenate files and print on the standard output

      0 - standard input
      1 - standard output
      2 - standard error

  cat > output.txt - redirection by default will remove(truncate) everything that
is in the file
  cat >> output.txt - if we want to keep the info that is currently in the file and
add new one(appending to file)
  cat 2>> error.txt - redirect all errors in error.txt
  cat >> output.txt 2>>error.txt
  cat < input.txt
```

tty - each terminal window or shell is a separate tty device. You can find its location via 'tty'

# Piping: The pipe is used to combine two or more commands, and in this, the output of one command acts as input to another command, and this command's output may act as input to the next command, and so on.

```
  date > date.txt
  cut < date.txt --delimiter " " --fields 1
```

OR with piping:

```
  date | cut --delimiter " " --fields 1
  date | cut --delimiter " " --fields 1 > today.txt
```

- Redirections are processed before pipes, which can brake the pipe(Tee command to be used)

```
  date | tee full_date.txt | cut --delimiter=" " --fields=1
```

- Turn your pipeline data into command line arguments(xargs)

```
  date | xargs echo

  rm deleteme.txt
  cat filestodelete.txt | xargs rm
```

# Aliases:

Search btn - Text editor

rename the file to .bash_aliases and save it - .makes the file a hidden file

```
  alias getdates='date | tee /home/bogomila/fulldate.txt |cut --delimiter=" " --fields=1 ] tee /home/bogomila/shortdate.txt | xargs echo hello'
  alias name_of_alias='..............'
  alias calmagic='xargs cal -A 1 -B > /home/bogomila/thing.txt'
  echo "12 2017" | calmagic
```