03. Command Basics

Commands table

command	**use**
`Alt + Ctrl + t`	to open a terminal in command

command	application
`date`	prints current date and time
`ncal`	prints current month calendar organized vertically.
`cal`	prints current month calendar organized horizontally

command	application
`ncal <year>`	prints calendar of the year passed.
`ncal <month> <year>`	prints calendar of the month of year passed.
`sort <file_name>`	sorts the context of the file and prints.
`rm <file_name>`	removes the file.

Command	Application
`ncal -h`	prints calendar without highlighting current date
`ncal -j`	prints calendar using Julian days
`nacl -M`	prints calendar using Monday as first day of the week
`nacl -3`	prints previous, current, next month calendar

command	Application
`ncal -h -j -3`	prints previous, current, next month calendar without highlighting current day in julian days

Commands	**Applications**
`date --universal`	prints present date n UTC format
`sort --reverse <file_name>`	prints reversed sorted content of from the given file
`sort --uinque <file_name>`	prints unique options form the given file

01. What actually matters in this Section

- not everything is critical to learn
- what is useful, important, and nice to have is shown at the beginning of the section

02. Opening Up Terminal

- use command `Alt + Ctrl + t` to open a terminal in command
- or simply fire up from the super key menu

03. Understanding the prompt

- When we open up our terminal, we’ll see our prompt which will likely include `username@machinename` followed by a `~` and `$` sign.

    - example: `pank@pank-VB:~$`

- while executing commands which may take some time like downloading packages, prompts may not be visible
- if we type random invalid texts as a command, Linux will look for that command and if it’s not in the list then it throws `command not found`.
- while writing commands for a case if we didn’t write commands without proper closing of the quotation or parenthesis then it will keep looking for that closing of the command and even valid commands will not get registered.

04. Our First commands!

- note: Linux commands are case sensitives. Sometimes it will suggest closest similar commands.
- For UNIX though, some commands are not case sensitives.

command	application
`date`	prints current date and time
`ncal`	prints current month calendar organized vertically.
`cal`	prints current month calendar organized horizontally

05. Using arrow keys in terminal

- `left` and `right` arrows are used to navigate currently written in current time.
- `up` and `down` arrows are used to navigate to history of commands. up arrow starts with most recent commands

06. Command Line Arguments

- most commands support multiple options that modify the behavior. Options are decided by us to include and execute.
- also many commands accept arguments (the things that the command acts upon or uses)
    - `arguments` are the values that we provide to the commands
- example: `ncal 2020` will print calendar of entire year 2020. Here 2020 is an argument passed to the command.

command	application
`ncal <year>`	prints calendar of the year passed.
`ncal <month> <year>`	prints calendar of the month of year passed.
`sort <file_name>`	sorts the context of the file and prints.
`rm <file_name>`	removes the file.

07. Providing Options To Commands

- options modify the behavior of the command in predefined ways
- options are prefixed by a dash `-`
    - example: `sort -r` is used to reverse sort the contents
- options are also case sensitives.

Command	Application
`ncal -h`	prints calendar without highlighting current date
`ncal -j`	prints calendar using Julian days
`nacl -M`	prints calendar using Monday as first day of the week
`nacl -3`	prints previous, current, next month calendar

08. Providing Multiple Options

- we can use multiple options at same time to get more customized results
- multiple command are chained together and makes the command work together
- options can be chained together with single `-` or multiple.

command	Application
`ncal -h -j -3`	prints previous, current, next month calendar without highlighting current day in julian days

09. Using Long Form Options

- some command can have both shorter and longer options to work with.
- to use longer commands, we use dual dash `--`
- example: `date -u` and `date --universal` both will print date in UTC

Commands	**Applications**
`date --universal`	prints present date n UTC format
`sort --reverse <file_name>`	prints reversed sorted content of from the given file
`sort --uinque <file_name>`	prints unique options form the given file

10. Options That Requires Parameters

- some options requires to pass parameters to it in order to print results
- example: `ncal -A 2` prints calendar of present month and also 2 months next from the present month. while `ncal -B 2` prints same for previous two months calendars and present month