GETTING STARTED
WITH
TEXT EDITORS

➢ TEXT EDITORS:

- Linux comes with several applications that can be used to write and edit program code.
- You can constantly need to edit config files, write scripts, change system files…all required to use a text editor.
- The most popular text editors are:
  - vi / vim editor
  - nano
  - gedit
  - emacs
  - kate / kwrite
  - lime text
  - pico edito

➢ VI / VIM EDITOR:

- Vi / Vim is a powerful **shell prompt text editor.** That means doesn't open a **GUI window.**
- It is most popular and widely used text editors among system administrators and programmers.
- It enables **syntax highlighting** when writing code or editing configuration files.
- Vi / VIM follows three modes. There are:
  - Command mode
  - Insert mode
  - Colon or execution mode

NOTE: The command mode is default.

SYNTAX: **$vim [arguments] filename**

→ **Create a new file using vi/vim:**

$vim filename

Press **"i"** to go insert mode.

Add some content then **esc, shift+:** to save a file.

KEYBOARD SHORTCUTS FOR COMMAND MODE:

- In Vi, you don't use your mouse or any of the keyboard shortcuts you're used to in other editors.
- For instance, you don't copy with Ctrl+C and paste with Ctrl+V because Vi has its own set of quick shortcuts that optimize key presses.

MODES:

i : For insert mode

I : Insert text at beginning of current line

a : Append text after cursor

A : Append text at end of current line

o : To insert a new line below the cursor position.
O : To insert a new line above the cursor position.

Esc : Command mode

NAVIGATION OPTIONS:

h : cursor left
j : cursor down
k : cursor up
l : cursor right
b : cursor to beginning of current word
e : cursor to end of current word
0 (zero) : go to beginning of line
$: Cursor goes to end of the current line
^ : Cursor goes to begging of current line
u : Undo whatever you did
r : Replace single character
gg (ngg) : Go to first line in a file / nth line of a file
G (nG) : Last line in file
v : select text
V : select line

COPY AND PASTE:

y	**:** copy ("yank") selection
yy (nyy)	**:** copy ("yank") the entire line / n lines to copy
p	**:** paste below the cursor
P	**:** Paste above the cursor

DELETE:

x	**:** delete character
nx	**:** delete n characters
dw	**:** delete word
dd (ndd)	**:** delete a line
d$	**:** delete from current position to end of line
d0	**:** delete from current position to beginning of line
dG	**:** delete to last line of file

UNDO:

u	**:** undo the last action
U	**:** undo all recent changes made to the current line
ctrl-r	**:** redo

SAVE & QUITTING:

:w	**:** save without quit.
:w!	**:** save with forcefully
:q	**:** quit without save
:q!	**:** force quit without saving
:wq	**:** save and quit
:wq! or :x	**:** save and quit with forcefully

SEARCHING:

/cloud	**:** search the file for "cloud". This will highlight all matches in the editor for you to see.
?cloud	**:** search the file for "cloud".
n / N	**:** find next / previous occurrence of search word

LINE NUMBERS:

:se nu	**:** Set line numbers
:se nonu	**:** Remove line numbers
:n	**:** go to nth line

READING COMMAND OUTPUT:

:! Pwd	**:** To print current working directory
:r !date	**:** To read date command output
:r !cat filename	**:** To read filename output
:1,4 co 8	**:** copy 1,4 lines and paste after 8th line
:1,4 mo 8	**:** move 1,4 lines and paste after 8th line
:1,4 w test1	**:** writing lines 1,4 into a test1 file.

FIND & REPLACE:

:%s/cat/tiger/	**:** replace string **tiger** for the first occurrence in line
:%s/cat/tiger/g	**:** replace string tiget for each instance in file
:%s/cat/tiger/gi	**:** ignore case sensitive
:%s/cat/tiger/gic	**:** ask for confirmation before replace

MANAGE MORE FILES:

→ To manage multiple files at a time:

$vim -o file1 file2

Options:

:n : edit next file (file2)

:rew : rewind to the file (file1)

:ctrl +w : cursor move one file to another

NOTE: Hold ctrl key Press **w (twice)**

→ File open with last line directly:

$vim + filename

→ File open +n th line:

$vim +n filename

$vim +5 filename

→ File open a readonly mode:

$vim -R filename

➢ NANO EDITOR:

- Text editing is essential to Linux users. Historically, the Vim text editor has been the default tool for managing file contents.
- Today, many systems and sysadmins prefer to use the nano text editor.
- Nano is quite a bit more intuitive than Vim, but it's still worth taking a brief look at its most fundamental features.
- You must be able to accomplish the following four tasks with nano:
  - Create/open a file
  - Edit the file
  - Save changes
  - Exit the file

SYNTAX: **$nano [options] <filename>**

→ To create or open a file:

$nano demofile

- The file may now be edited. Nano does not use the concept of modes like Vim. Once the file opens in nano, if you type on the keyboard, text is inserted into the file.
- Use the same command syntax to open an existing file, such as the demofile.

SAVE A FILE:

- The Ctrl key activates nano's commands on your keyboard.
- There is an abbreviated list of commands displayed at the bottom of the nano interface, and these cover most basic needs.
- You **"write out"** a file to save its contents by using **Ctrl+O**.

FINDING YOUR LOCATION:

- Need to find out where your cursor is in the document? The **Ctrl+C** shortcut provides the line, column, character, and percent of the way through the document.

EXIT NANO:

- If you examine the list of available commands at the bottom of nano's interface, you discover that you won't "quit" nano, but instead "exit" nano by using **Ctrl+X**.