

Text Editors: vi/vim

Session 10

Text Editors: vi/vim

Session 10

Objectives:

- Understand the purpose and value of CLI text editors
- Use basic and advanced navigation in Vim
- Edit, copy, delete, and manipulate blocks of text using Vim modes
- Perform searches and global substitutions
- Choose between Vim and Emacs depending on context

What's a Text Editor? Why Use CLI Tools?

Content:

- Not word processors (no formatting)
- Fast, portable, no GUI needed
- Used for configs, logs, scripts, coding

What is VI/VIM?

Content:

- A screen-oriented text editor
- Released in 1991
- Improved version of “vi” released in 1976
- A “programmer’s editor” ● No mouse or ‘GUI’

Why is VI/VIM?

- **Efficiency:** Keyboard-only = very fast
- **Ergonomics:** Vim is designed to have your fingers always stay near the home row. This reduces hands fatigue and improves performance
- **Always available** (vim is POSIX)
- **It's Everywhere!** Vim exists on almost all machines. It's heavily used for system administration, programming, working with markup languages, and more!

VIM modes

vim is known for a modal editing scheme.

What is modal editing? In vim, various things that you typically do while editing text is split into different modes.

Normal mode (Esc)

Editor commands are used in this mode. The default mode.

Command-line mode (:)

Single line input, used mostly for file navigation. Exits mode after command

Insert mode (i)

Editing a text buffer. Similar to modern text editors.

Select mode (gh)

Printable characters replace text and Vim enters insert mode.

Visual mode (v)

Used for highlighting areas for text.

Ex mode (Q)

Similar to command-line mode, but does not exit mode after command

VIM modes

Mode	Purpose	Enter With	Example / Behavior
Normal	Move, delete, copy, paste, etc.	Press Esc	dd deletes a line, yy copies a line, p pastes below
Insert	Type text like in any editor	i, a, o, O	i – insert <i>before</i> cursor a – insert <i>after</i> cursor o – new line <i>below</i> O – new line <i>above</i>
Visual	Select text (char/line/block)	v, V, Ctrl+v	v – select chars, then y to copy V – select whole lines Ctrl+v – block/column
Command	Save, quit, search, replace	: (in Normal mode)	:w to save, :q! to quit without saving, :%s/foo/bar/g to replace text globally

Normal mode: Navigation

So how do we move around without a mouse?



Normal mode: Navigation

So how do we move around without a mouse?

Command

h j k l

w, b, e

0, ^, \$

gg, G

%

What It Does

Move left/down/up/right

Move to next, previous, end of word

Start of line, first char, end

Go to top/bottom of file

Jump to matching (), {}

Additionally:

👉 Use numbers before motions to repeat:

5j → move down 5 lines

3w → skip 3 words

Search and Replace in Vim

Command

/word

?word

n

N

:%s/foo/bar/g

:%s/foo/bar/gc

Description

Search forward for the word "word"

Search backward for the word "word"

Jump to the next match

Jump to the previous match

Replace all "foo" with "bar" in the file

Replace with confirmation (y/n/a/q)

Copy paste in VIM

Command	Action	Description
yy	Yank (copy) current line	Use p to paste after, P to paste before
y\$	Yank to end of line	Copy from cursor to end
yw	Yank one word	Copy one word from the cursor
p	Paste after cursor	Common after yy or yw
P	Paste before cursor	Useful in visual mode
dd	Delete current line	Also stores line in register (can paste later)
dw	Delete one word	From cursor to start of next word
d\$	Delete to end of line	From cursor to end
x	Delete one character under cursor	Simple char deletion
u	Undo last change	Step-by-step undo
Ctrl+r	Redo last undone change	Opposite of u

Live demo

- Create a file:
 - vim demo.txt

You are now in **Normal Mode** (default on launch), which is for navigating and controlling the editor.

- Press Esc to ensure Normal Mode.
- Type i to be in the **Insert Mode**, type Hello and then <Esc> to come back to the **Normal Mode**
- Now we will try:
 - dd " delete current line
 - u " undo
 - Ctrl+r " redo.
- Write "Hello!" In the file.
- Then try:
 - iBEFORE → Esc
→ Result: BEFORE Hello!
 - Move to ! (\$), press a AFTER → Esc
→ Result: BEFORE Hello! AFTER
 - oThis is a new line below. → Esc
 - OThis is a new line above. → Esc

This is a new line above.
BEFORE Hello! AFTER
This is a new line below.

Live demo

Visual Mode (v) Use Visual Mode to select and operate on blocks.

v → use l to select → y to yank

V → select line → d to delete

Ctrl+v → block selection → > to indent

Practice:

- v → move to select Hello → press y to yank it
- Move cursor to BEFORE → press p to paste after

Result: BEFOREHello Hello! AFTER

- V → select full line → press y
- Move below → press p to paste copied line
- Use Ctrl+v → select 3 columns across 2 lines → press > to indent

- :w " Save
- :q! " Quit without saving
- :wq " Save and quit

**This is a new line above. BEFOREHello
Hello! AFTERThis is a new line below.
BEFOREHello Hello! AFTER**

Live demo

Visual Mode (v) Use Visual Mode to select and operate on blocks.

v → use l to select → y to yank

V → select line → d to delete

Ctrl+v → block selection → > to indent

Practice:

- v → move to select Hello → press y to yank it
- Move cursor to BEFORE → press p to paste after

Result: BEFOREHello Hello! AFTER

- V → select full line → press y
- Move below → press p to paste copied line
- Use Ctrl+v → select 3 columns across 2 lines → press > to indent

- :w " Save
- :q! " Quit without saving
- :wq " Save and quit

**This is a new line above. BEFOREHello
Hello! AFTERThis is a new line below.
BEFOREHello Hello! AFTER**