

# VIM Editor

## Ubuntu (Debian-based)

```
sudo apt update
sudo apt install vim
```

apt update: Updates the package list.  
apt install vim: Installs Vim.

## CentOS (RHEL-based)

**For CentOS 7 or CentOS Stream 8/9:**

```
sudo dnf install vim
```

If you're using CentOS 7 and dnf is not available:

```
sudo yum install vim
```

### Notes:

To check if Vim is already installed:

```
vim --version
```

```
vagrant@vagrant:~$ vim --version
```

```
=====
VIM - Vi IMproved 8.0 (2016 Sep 12, compiled Jun 06 2019 17:31:41)
Included patches: 1-1453
```

```
...
```

If you want a minimal version (vim-minimal) is usually preinstalled, but lacks features.

For full-feature Vim:

```
sudo apt install vim
```

# or

```
sudo dnf install vim-enhanced
```

# Create, Edit and Navigate a File with VIM (Ubuntu)

```
vagrant@vagrant:~$ vim testfile.txt
```

```
=====
insert test
```

```
~
```

```
~
```

```
~
```

```
"testfile.txt" [New File]
```

## VIM Modes:

### 1. Normal Mode (Command Mode)

**Purpose:** Navigate, delete, copy, paste, and execute commands.

**How to enter:** Press Esc

**Examples:**

dd → delete a line

yy → yank (copy) a line

p → paste

u → undo

/text → search for "text"

### 2. Insert Mode

**Purpose:** Insert or edit text.

**How to enter:**

i → insert before the cursor

a → append after the cursor

o → open a new line below

**How to exit:** Press Esc to return to Normal mode.

### 3. Command-Line Mode (Colon Mode)

**Purpose:** Save, quit, search, and execute advanced commands.

**How to enter:** Press : from Normal Mode

### Examples:

:w → save (write)

:q → quit

:wq → save and quit

:q! → force quit without saving

## Optional: Other Modes (for advanced users)

**Visual Mode (v, V, Ctrl+v):** For selecting text.

**Replace Mode (R):** Overwrite characters directly.

## Essential Vim shortcuts/commands:

```
vagrant@vagrant:~$ vim testfile.txt
```

```
=====
```

```
1 insert test
```

```
2
```

```
~
```

```
~
```

```
:set number
```

```
=====
```

## General Mode Switching

Key	Action
Esc	Switch to <b>Normal mode</b> from Insert/Visual
i	Enter <b>Insert mode</b> before the cursor
a	Enter Insert mode <b>after</b> the cursor
v	Enter <b>Visual mode</b> (character selection)
:	Enter <b>Command-line mode</b>
R	Enter <b>Replace mode</b>

## Navigation (Normal Mode)

Key	Action
h	Left
j	Down
k	Up
l	Right
w	Jump to next word

b	Jump to previous word
0	Jump to beginning of line
^	Jump to first non-blank character
\$	Jump to end of line
gg	Go to top of file
G	Go to bottom of file
:n	Go to line n (e.g. :42)

## Editing (Normal Mode)

Key	Action
x	Delete character under cursor
dd	Delete entire line
dw	Delete word
yy	Yank (copy) a line
p	Paste after cursor
P	Paste before cursor
u	Undo
Ctrl+r	Redo
r<char>	Replace character under cursor with <char>
~	Toggle character case

## Searching

Key	Action
/text	Search <b>forward</b> for "text"
?text	Search <b>backward</b> for "text"
n	Repeat last search forward
N	Repeat last search backward

## Saving and Exiting (Command-Line Mode)

Command	Action
:w	Save (write)
:q	Quit
:wq	Save and quit
:q!	Quit without saving
:x	Save and exit (like :wq)

**Visual Mode**

Key	Action
v	Start character-wise selection
V	Start line-wise selection
Ctrl+v	Start block/column selection
y	Yank (copy) selection
d	Delete selection
> / <	Indent / un-indent selection

# Search and Replace

## Basic Substitution Syntax:

`:[range]s/old/new/[flags]`

## Common Example: Substitute "foo" with "bar" in the entire file

`:%s/foo/bar/g`

## Breakdown:

`:` — Enters command-line mode

`%` — Applies to the **entire file**

`s` — Substitute command

`foo` — The **search term**

`bar` — The **replacement string**

`g` — Global flag (replaces **all** occurrences in a line, not just the first)

## Other Useful Variants:

### Replace only the first occurrence in each line:

`:%s/foo/bar/`

### Replace with confirmation:

`:%s/foo/bar/gc`

`c` — Confirm each replacement (y to replace, n to skip)

### Replace only in a specific range (e.g., lines 10 to 20):

`:10,20s/foo/bar/g`

```
vagrant@vagrant:~$ vim testfile.txt
```

```
""  
two  
three  
four  
two  
ten
```

```
~  
/two
```

```
vagrant@vagrant:~$ vim testfile.txt
```

```
""  
NONE  
three  
four  
NONE  
ten
```

```
~  
~  
:%s/two/NONE/
```

```
vagrant@vagrant:~$ vim testfile.txt
```

```
""  
NONE  
three  
four  
NONE  
ten
```

```
~  
~  
replace with two (y/n/a/q/l/^E/^Y)? Y
```

```
vagrant@vagrant:~$ vim testfile.txt
```

```
""  
two  
three  
four  
NONE  
ten
```

```
~  
~  
replace with two (y/n/a/q/l/^E/^Y)?
```

