

vi/vim text editor

vi/vim text editor

- Advanced text editors
 - vi/vim
 - emacs

```
VIM - Vi IMproved

        version 8.2.4919
        by Bram Moolenaar et al.
    Modified by team+vim@tracker.debian.org
Vim is open source and freely distributable


        Sponsor Vim development!

type  :help sponsor<Enter>      for information

type  :q<Enter>                  to exit
type  :help<Enter> or <F1>      for on-line help
type  :help version8<Enter>    for version info
```

vi/vim text editor

- Vim is a powerful and efficient *text editor* that offers a vast array of features for editing and manipulating text.
- Vi stands for Vlsual editor
- Derived from its predecessor, Vi, Vim (Vi IMproved) has become a favorite among developers, system administrators, and writers due to its versatility and extensibility.

vi/vim Basic modes

1. Normal Mode: Default mode for navigation, editing, and executing commands.
2. Insert Mode: Text insertion and editing mode (Press i, a, o, etc.).
3. Visual Mode: Text selection mode (Press v to enter).
4. Command-Line Mode: Enter complex commands with :.
5. Replace Mode: Overwrite existing text (Press R).

Getting started

- **i** - Insert Mode
 - Transitions Vim to the insert mode for text input.
- **Esc** - Normal Mode
 - Returns Vim to the normal mode
- **:w** - Write
 - Saves changes made to the file.
- **:q** - Quit
 - Exits Vim.
- **:wq** - Write and Quit
 - Saves changes and exits Vim.

vi/vim - open vim

- **vi +100 foo**: Open the file "foo" and position the cursor at line 100.
- **vi +/pat foo**: Open the file "foo" and position the cursor at the first occurrence of the pattern "pat."
- **vi + foo**: Open the file "foo" and position the cursor at the end of the file.
- **vi -R foo**: Open the file "foo" in read-only mode, preventing any changes to the file.
- **vi -O file1 file2**: Open multiple files ("file1" and "file2") in vertical split mode.
- **vi -o file1 file2**: Open files in horizontal split mode.

vi/vim - Saving and Quitting

- `:w` - Save file and remain in editing mode.
- `:w bar` - Save the file as "bar" (similar to "Save As" in Windows).
- `:w! bar` - Overwrite the existing file "bar" without confirmation.
- `:n1,n2w foo` - Write lines from `n1` to `n2` to the file named "foo".
- `:n1,n2w>> foo` - Append lines from `n1` to `n2` to the file named "foo".
- `:x` - Save the file and quit editing mode (same as `:wq`).
- `:q` - Quit editing mode when no changes are made to the file.
- `:q!` - Quit editing mode and abandon changes without confirmation.

vi/vim - Copying

- **yy**: Copy the current line.
- **6yy**: Copy six lines.
- **y**: Copy the selected region (Vim only).
- **p**: Paste the copied text on the right.
- **P**: Paste the copied text on the left or above.

vi/vim - Deleting

- **x**: Delete the character under the cursor.
- **6x**: Delete the character under the cursor and the five characters to the right.
- **X**: Delete the previous character (to the left of the cursor).
- **dd**: Delete the current line.
- **4dd**: Delete four lines from the current line.
- **dw**: Delete the word under the cursor.
- **d**: Delete the selected region (Vim only).
- **J**: Join the current line with the next line.
- **kJ**: Join the current line with the previous line.

vi/vim - Insert text

- **i**: Insert text to the left of the cursor.
- **I**: Insert text at the beginning of the line.
- **[Ctrl-v][Ctrl-b]**: Insert the text "[Ctrl-b]" (literal control characters).
- **[Ctrl-v][Esc]**: Insert the Escape key (literal escape character).
- **a**: Append text to the right of the cursor.
- **A**: Append text at the end of the line.
- **o**: Open a new line below the current line and enter insert mode.
- **O**: Open a new line above the current line and enter insert mode.

vi/vim - Interface to Linux

- `:!cmd` - Run a UNIX command `cmd`.
 - Example: `:!ls` - Execute the `ls` command in the shell.
- `:%` - Execute the current file as a shell or Perl script.
- `:r !cmd` - Read in the output of command `cmd` below the current line.
 - Example: `:r !head -n 3 foo` - Read the first three lines of "foo" below the current line.
- `:sh` or [Ctrl-z] - Escape to the UNIX shell temporarily (suspend editor).

vi/vim - Undo and Redo

- **u** - Undo last editing command
- **[Ctrl-r]** - Redo last undo (vim only)
- **U** - Undo all changes made to current line

vi/vim - Substitution

`:[range]s/pattern/replacement/[flags]`

- `[range]`: Specifies the range of lines to perform the substitution. If omitted, the substitution will be applied to the current line only.
- `pattern`: The text pattern to search for (can be a regular expression).
- `replacement`: The text to replace the matched pattern.
- `flags`: Optional flags that modify the behavior of the substitution

Examples

`1,$s/s1/s2/g` - Replace string "s1" with "s2" globally in all lines from 1 to the end.

`1,$s/s1/s2/gc` - Interactive replacement - asks for confirmation for each match.

vi/vim - Search and Repeat

- `/pat`: Non-incremental search forward for the string "pat."
- `?pat`: Non-incremental search backward for the string "pat."
- `n`: Repeat the last string search in the same/forward direction.
- `N`: Repeat the last string search in the opposite/backward direction.
- `:set ic`: Ignore case when searching.

vimtutor

Takes about 30 min to
complete the tutorial

```
=====
=  Welcome to the VIM Tutor - Version 1.7  =
=====
```

Vim is a very powerful editor that has many commands, too many to explain in a tutor such as this. This tutor is designed to describe enough of the commands that you will be able to easily use Vim as an all-purpose editor.

The approximate time required to complete the tutor is 25-30 minutes, depending upon how much time is spent with experimentation.

ATTENTION:

The commands in the lessons will modify the text. Make a copy of this file to practice on (if you started "vimtutor" this is already a copy).

It is important to remember that this tutor is set up to teach by use. That means that you need to execute the commands to learn them properly. If you only read the text, you will forget the commands!

Now, make sure that your Shift-Lock key is NOT depressed and press the `j` key enough times to move the cursor so that Lesson 1.1 completely fills the screen.

```
~~~~~
Lesson 1.1:  MOVING THE CURSOR
```

References

1. Your Unix :The Ultimate Guide, Das, Sumitabha