**How to exit the Vim editor?**

**Mehod 1:**

Hit the Esc key to enter "Normal mode". Then you can type : to enter "Command-line mode". A colon (:) will appear at the bottom of the screen and you can type in one of the following commands. To execute a command, press the Enter key.

* :q to quit (short for :quit)
* :q! to quit without saving (short for :quit!)
* :wq to write and quit
* :wq! to write and quit even if file has only read permission (if file does not have write permission: force write)
* :x to write and quit (similar to :wq, but only write if there are changes)
* :exit to write and exit (same as :x)
* :qa to quit all (short for :quitall)
* :cq to quit without saving and make Vim return non-zero error (i.e. exit with error)

You can also exit Vim directly from "Command mode" by typing ZZ to save and quit (same as :x) or ZQ to just quit (same as :q!). (Note that case is important here. ZZ and zz do not mean the same thing.)

Vim has extensive help - that you can access with the :help command - where you can find answers to all your questions and a tutorial for beginners.

**Method 2:**

Before you enter a command, **hit the Esc key**. After you enter it, hit the Return to confirm.

Esc finishes the current command and switches Vim to [normal mode](http://vimdoc.sourceforge.net/htmldoc/intro.html#Normal). Now if you press :, the : will appear at the *bottom* of the screen. This confirms that you're actually typing a command and not editing the file.

Most commands have abbreviations, with optional part enclosed in brackets: c[ommand].

Commands marked with '\*' are Vim-only (not implemented in Vi).

**Safe-quit (fails if there are unsaved changes):**

* :q[uit] Quit the current [window](http://vimdoc.sourceforge.net/htmldoc/windows.html#window). Quit Vim if this is the last window. This fails when changes have been made in current [buffer](http://vimdoc.sourceforge.net/htmldoc/windows.html#buffers).
* :qa[ll]\* Quit all windows and Vim, unless there are some buffers which have been changed.

**Prompt-quit (prompts if there are unsaved changes)**

* :conf[irm] q[uit]\* Quit, but give prompt when there are some buffers which have been changed.
* :conf[irm] xa[ll]\* Write all changed buffers and exit Vim. Bring up a prompt when some buffers cannot be written.

**Write (save) changes and quit:**

* :wq Write the current file (even if it was not changed) and quit. Writing fails when the file is read-only or the buffer does not have a name. :wqa[ll]\* for all windows.
* :wq! The same, but writes even read-only files. :wqa[ll]!\* for all windows.
* :x[it], ZZ(with [details](http://vimdoc.sourceforge.net/htmldoc/editing.html#ZZ)). Write the file only *if it was changed* and quit, :xa[ll]\* for all windows.

**Discard changes and quit:**

* :q[uit]! ZQ\* Quit without writing, also when visible buffers have changes. Does not exit when there are changed hidden buffers.
* :qa[ll]!\*, :quita[ll][!]\* Quit Vim, all changes to the buffers (including hidden) are lost.

**Press Return to confirm the command.**

This answer doesn't reference all Vim write and quit commands and arguments. Indeed, they are referenced in the [Vim documentation](http://vimdoc.sourceforge.net/htmldoc/editing.html#:q).

Vim has extensive built-in help, type Esc:helpReturn to open it.

**Method 3:**

VIM has 3 modes of operation: Input mode, Command mode & Ex mode.

**Input mode** - everything that you type, all keystrokes are echoed on the screen.

**Command mode or Escape mode** - everything that you type in this mode is interpreted as a command.

**Ex mode** - this is another editor, ex. It is a line editor. It works per line or based on a range of lines. In this mode, a : appears at the bottom of the screen. This is the ex editor.

In order to exit vim, you can exit while you are in either the ex mode or in the command mode. You cannot exit vim when you are in input mode.

**Exiting from ex mode**

1. you need to be sure that you are in the Command mode. To do that, simply press the Escape key.
2. Go to the ex mode by pressing the : key
3. Use any of the following combinations in ex mode to exit:

:q - quit :q! - quit without saving :wq - save & quit or write & quit :wq! - same as wq, but force write incase file permissions are readonly :x - write & quit :qa - quit all. useful when multiple files are opened like: vim abc.txt xyz.txt

**Exiting from command mode**

1. Press the escape key. You probably have done this already if you are in command mode.
2. press capital ZZ (shift zz) - save & exit
3. press capital ZQ (shift zq) - exit without saving.

**Vim is an editor to create or edit a text file.**

There are two modes in vim. One is the command mode and another is the insert mode.

In the command mode, user can move around the file, delete text, etc.

In the insert mode, user can insert text.

**Changing mode from one to another**

**From** command mode **to** insert mode type **a**/**A**/**i**/**I**/**o**/**O** ( see details below)

**From** insert mode **to** command mode type **Esc** (escape key)

Some useful commands for VIM

**Text Entry Commands** (Used to start text entry)

**a** Append text following current cursor position

**A** Append text to the end of current line

**i** Insert text before the current cursor position

**I** Insert text at the beginning of the cursor line

**o** Open up a new line following the current line and add text there

**O** Open up a new line in front of the current line and add text there

**The following commands are used only in the commands mode.**

**Cursor Movement Commands**

h Moves the cursor one character to the left

l Moves the cursor one character to the right

k Moves the cursor up one line

j Moves the cursor down one line

nG or :n Cursor goes to the specified (n) line

*(ex. 10G goes to line 10)*

^F (CTRl F) Forward screenful

^B Backward screenful

^f One page forward

^b One page backward

^U Up half screenful

^D Down half screenful

$ Move cursor to the end of current line

0 (zero) Move cursor to the beginning of current line

w Forward one word

b Backward one word

**Exit Commands**

:wq Write file to disk and quit the editor

:q! Quit (no warning)

:q Quit (a warning is printed if a modified file has not been saved)

ZZ Save workspace and quit the editor (same as :wq)

: 10,25 w temp

write lines 10 through 25 into file named temp. Of course, other line

numbers can be used. (Use :f to find out the line numbers you want.

**Text Deletion Commands**

x Delete character

dw Delete word from cursor on

db Delete word backward

dd Delete line

d$ Delete to end of line

d^ (d caret, not CTRL d) Delete to beginning of line

**Yank** (has most of the options of delete)-- VI's **copy** commmand

yy yank current line

y$ yank to end of current line from cursor

yw yank from cursor to end of current word

5yy yank, for example, 5 lines

**Paste** (used after delete or yank to recover lines.)

p paste below cursor

P paste above cursor

"2p paste from buffer 2 (there are 9)

u Undo last change

U Restore line

J Join next line down to the end of the current line

**File Manipulation Commands**

:w Write workspace to original file

:w file Write workspace to named file

:e file Start editing a new file

:r file Read contents of a file to the workspace

**To create a page break,** while in the insert mode, press the CTRL key

And l. ^L will appear in your text and will cause the printer to start

A new page.

**Other Useful Commands**

Most commands can be repeated n times by typing a number, n, before

the command. For example 10dd means delete 10 lines.

. Repeat last command

cw Change current word to a new word

r Replace one character at the cursor position

R Begin overstrike or replace mode � use ESC key to exit

:/ pattern Search forward for the pattern

:? pattern Search backward for the pattern

n (used after either of the 2 search commands above to

continue to find next occurrence of the pattern.

:g/pat1/s//pat2/g replace every occurrence of pattern1 (pat1) with

pat2

**Example :**g/tIO/s//Ada.Text\_IO/g

This will find and replace tIO by Ada.text\_IO everywhere in the file.

:g/a/s// /g replace the letter a, by blank

:g/a/s///g replace a by nothing

note: Even this command be undone by u

**Examples**

**Opening a New File**

Step 1 type **vim *filename*** (create a file named *filename*)

Step 2 type **i** ( switch to insert mode)

Step 3 enter text (enter your Ada program)

Step 4 hit **Esc key** (switch back to command mode)

Step 5 type **:wq** (write file and exit vim)

**Editing the Existing File**

Step 1 type **vim *filename*** (edit the existing file named filename)

Step 2 move around the file using **h**/**j**/**k**/**l** key or any appropriate command

**h** Moves the cursor one character to the left

**l** Moves the cursor one character to the right

**k** Moves the cursor up one line

**j** Moves the cursor down one line

**nG** or :n Cursor goes to the specified (n) line

*(ex. 10G goes to line 10)*

Step 3 **edit** required text (replace or delete or insert)

Step 4 hit **Esc** key (exit from insert mode if you insert or replace text)

Step 5 type **:wq**