

# Vi IMproved (Vim) Quick Start Guide

Vim is a modal text editor used from either a command-line interface or within a graphical user interface (GUI).<sup>1</sup>

This document provides descriptions and procedural examples of using basic Vim commands, including explanations of opening and exiting the application, movement of the cursor, saving files, editing text, and searching for content. It provides information on accessing system help and supplies a list of resources outside of this introduction that delves deeper into all things Vim.

## Open Vim

There are multiple commands that can be used to open Vim. For this introductory tutorial, we will use the Vim command to open the editor in normal mode. [1]

To open Vim and clear the start screen to prepare for the addition of text:

1. Type *Vim* on your system's command line and press **Enter** as shown in Figure 1:

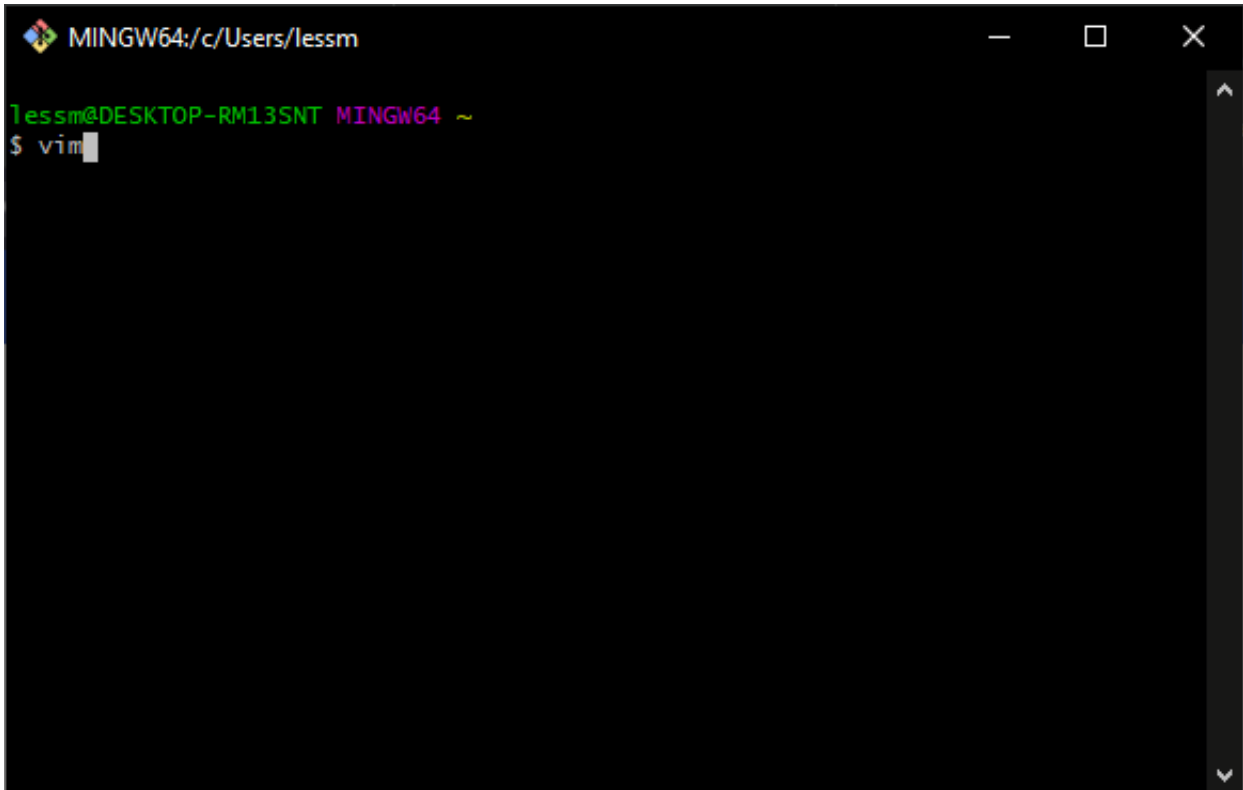


Figure 1. Open Vim from your system's command line.

---

<sup>1</sup> This document introduces entry-level system administrators and new Linux users to some of the most frequently used commands needed when working with files and content in Vim.

The action in Figure 1 displays a start screen in Vim similar to Figure 2:

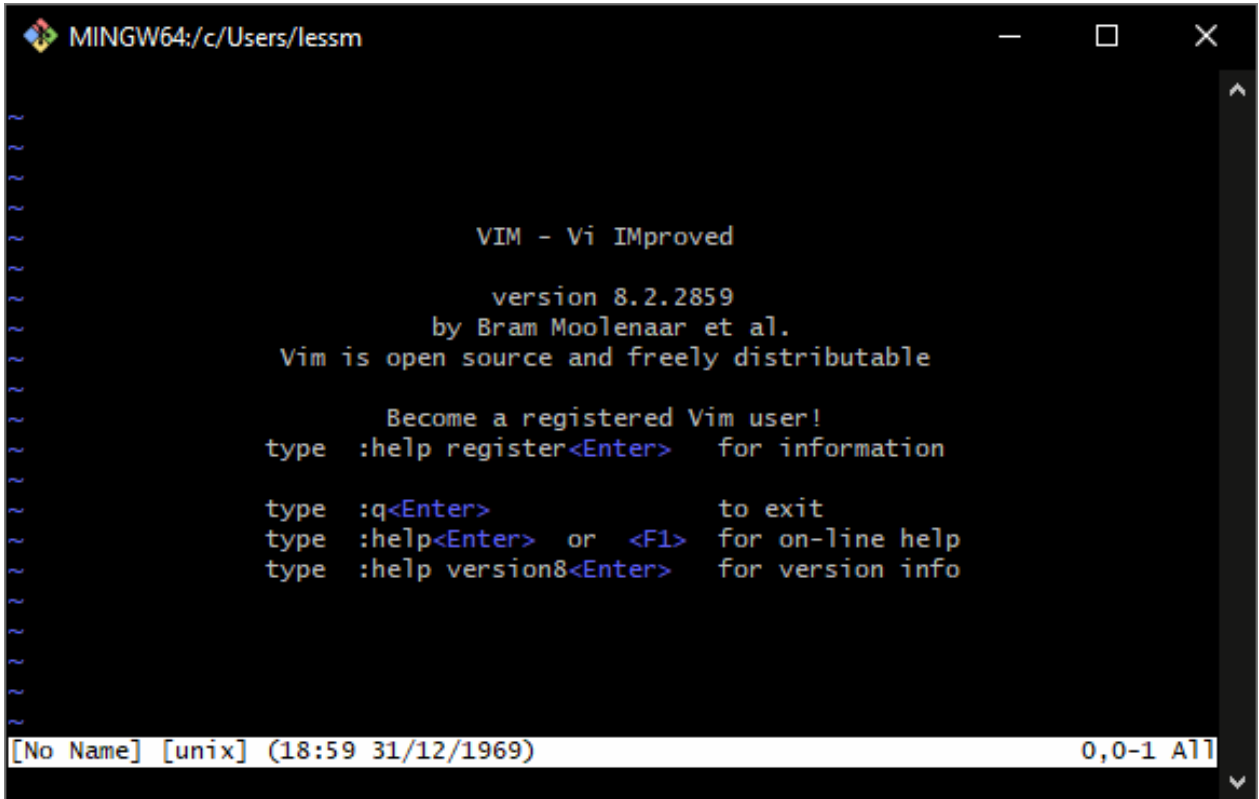


Figure 2. Vim start screen that displays after invoking Vim from the command line.

2. Press **Ctrl + I** to clear the text from the start screen to begin using the editor. [2]

## File Commands

Once you invoke Vim, you will want to learn some basic file commands necessary to create new files, save changes to files, and quit Vim while saving or discarding changes.

Table 1 contains a list of file commands, along with a description of the function for each:

Command	Function
Vim <i>filename</i>	Either creates and opens a new file or opens an existing file for edit outside of the Vim shell.
:e <i>filename</i>	Either creates and opens a new file or opens an existing file from the normal mode in Vim.
:w	Saves a file without leaving Vim.
:wq	Saves a file and quits Vim.
:q!	Discards any unsaved changes and quits Vim.

Table 1. Basic Vim File Commands [3]

## Create New and Open Existing Files

Unlike the other commands in Table 1, use the *Vim filename* command in the operating system's command line outside of the editor's shell. This command and the `:e` command both require the inclusion of a name for the new or existing file in its syntax (refer to Figure 3 to view an example). Vim searches the directory, and if a file by the specified name exists, it opens the file in the editor. If the file name does not exist in the directory, a new file is created. [3]

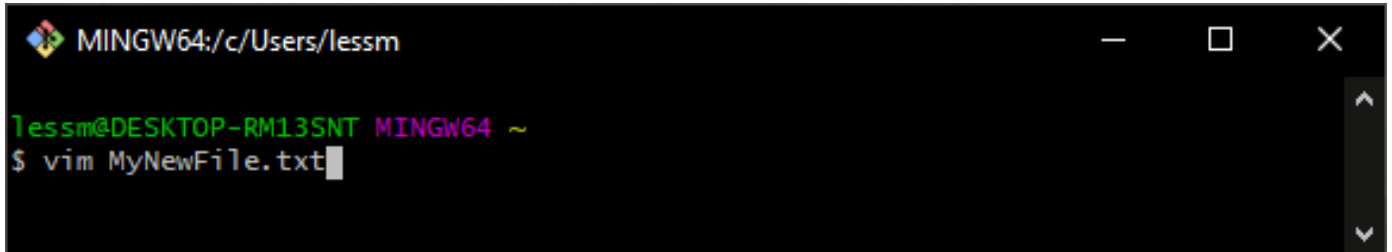


Figure 3. The Vim filename command creates a new file or opens an existing file.

When a new file is created, the name of the file appears in quotes beneath the status bar, along with the word "New" in brackets, as shown in Figure 4:



Figure 4. The new file message displays beneath the status bar.

*Note:* If the file already exists, only the file name displays beneath the status bar.

## Save and Discard Changes and Quit Vim

Perform these commands in **normal mode** to avoid disabling the command and entering it as text in the document.

To use file commands to save a file and remain in Vim or to save and quit Vim:

1. Press the **Esc** key to ensure Vim is in normal mode.
2. Type the *command* on the command line and press **Enter**. Refer to Table 1 to view a list of commands.

If executing the `:w` command, a message displays in the status bar as shown in Figure 5 to indicate the changes have been written to the file:

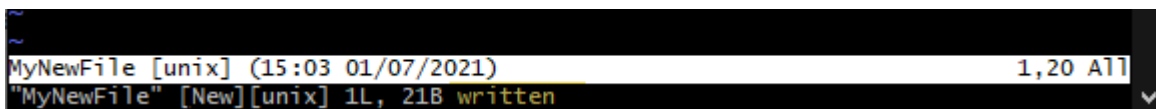


Figure 5. Vim returns a message to confirm the save action.

## Normal and Insert Modes

There are two modes in Vim every user needs to learn. Table 2 describes these modes:

Mode	Description
Normal	The default mode in Vim, in which users reposition the cursor and execute commands. Text cannot be entered while in this mode. To return to normal mode from a different mode (such as the insert mode), press the <b>Esc</b> key on the keyboard.
Insert	A mode that allows users to insert text. Commands cannot be executed while in this mode. To switch to insert mode from the normal mode, press the <b>i</b> key.

Table 2. Essential Modes for Use in Vim [4]

To activate insert mode, insert text, and return to normal mode:

1. Press **i** on the keyboard. The `INSERT` indicator displays beneath the status bar:

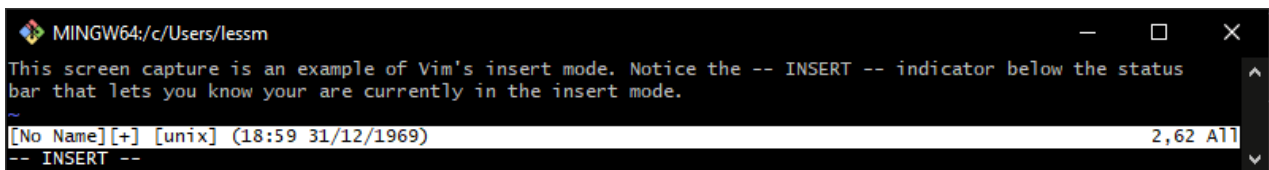


Figure 6. Vim session set to insert mode.

2. Begin typing to add text to the file.
3. Press the **Esc** key to return to normal mode and notice that in Figure 7, the `INSERT` indicator is no longer visible from the screen:

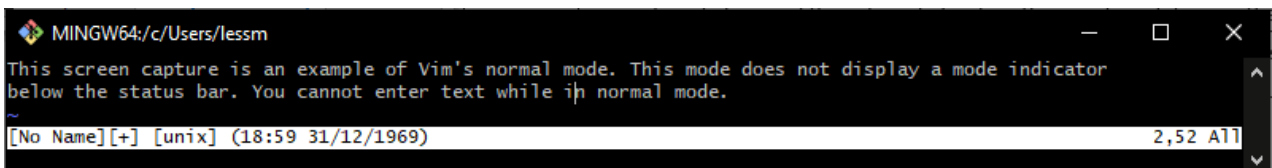


Figure 7. Vim session set to normal mode.

## Cursor Keys

Table 2 describes four cursor keys that enable users to move the cursor (up, down, left, right) directly from the keyboard when Vim is in `normal mode`:

Key	Direction	Function
h	←	Moves the cursor one space to the left of the current position.
j	↓	Moves the cursor down one line from the current position.
k	↑	Moves the cursor up one line from the current position
l	→	Moves the cursor one space right of the current position.

Table 3. Keys that Control Cursor Movement

The location of these cursor keys allows you to control the movement of the cursor without moving your right hand from the primary key set, thereby increasing overall speed.

Alternatively, you can use the directional arrow keys on the keyboard to move the cursor. However, this method decreases the rate of text entry speed because you must move your right hand from the primary key set over to the directional arrow keys and then back to the primary key set. [1]

## Vim Commands

Vim responds to many types of commands, such as those that control cursor movement, text edits, and searches. The following subtopics describe just a few of these commands to get you started with using Vim. Refer to [Additional Resources](#) to find sources that contain more information on Vim commands.

*Note:* Commands in Vim are case-sensitive.

### Cursor Movement Commands

Use cursor commands to move through words, lines of text, and blank lines (paragraphs) in Vim.

Perform cursor movement commands in [normal mode](#) to avoid disabling the command and entering it as text in the document. Table 3 contains a list of cursor movement commands, along with a description of the function for each.

Command	Motion
gg	Jumps cursor to the beginning of a file.
G	Jumps cursor to the end of a file.
0 (zero)	Jumps cursor to the beginning of a line.
\$	Jumps cursor to the end of a line.
w	Jumps the cursor forward from its current position to the beginning of the next word.
e	Jumps the cursor forward from its current position to the end of the next word.
b	Jumps the cursor backwards from its current position to the beginning of the previous word.
}	Jumps forward to the first blank line in the editor.
{	Jumps backward to the first blank line in the editor.

Table 4. Frequently Used Cursor Movement Commands [4]

To use a cursor command:

1. Press the **Esc** key to ensure Vim is in normal mode.
2. Enter the *command* from the keyboard. The application moves or jumps the cursor to its new location. Refer to [Table 4](#) to view a list of cursor commands.

## Text Commands

Use text commands to insert and delete text and undo and redo text changes.

Perform text commands in [normal mode](#) to avoid disabling the command and entering it as text in the document. Table 4 contains a list of text commands to use in Vim, along with a description of the function for each:

Command	Function
i	Activates <a href="#">insert mode</a> .
a	Activates insert mode and appends text directly after the cursor position.
o	Activates insert mode and opens a new line below the cursor position.
O	Activates insert mode and opens a new line above the cursor position.
x	Deletes a single character under the cursor.
dd	Deletes an entire line.
d\$	Deletes to the end of a line.
u	Undoes a single change.
U	Undoes all changes on the last edited line.
Ctrl + r	Reinstates undone changes.

Table 5. Frequently Used Text Commands [4]

To use a text command:

1. Press the **Esc** key to ensure Vim is in normal mode.
2. Enter the *command* from the keyboard. Vim performs the action that corresponds to the command. Refer to [Table 5](#) to view a list of text commands.

## Search Commands

Use search commands to locate words and phrases in text.

Perform search commands in [normal mode](#) to avoid disabling the command and entering it as text in the document.

*Note:* Search is case-sensitive.

Table 6 contains a list of search commands, along with a description of the function for each:

Command	Function
<i>/phrase</i>	Searches downward from the cursor position to the first occurrence of a word or phrase.
<i>?phrase</i>	Searches upward from the cursor position to the first occurrence of a word or phrase.
n	Repeats the search to locate the next occurrence of a word or phrase.
N	Repeats the search in the opposite direction to locate the next occurrence of a word or phrase.

Table 6. Frequently Used Search Commands [4]

To perform a search:

1. Press the **Esc** key to ensure Vim is in normal mode.
2. Type */phrase* on the command line. Refer to [Table 6](#) to view a list of text commands.

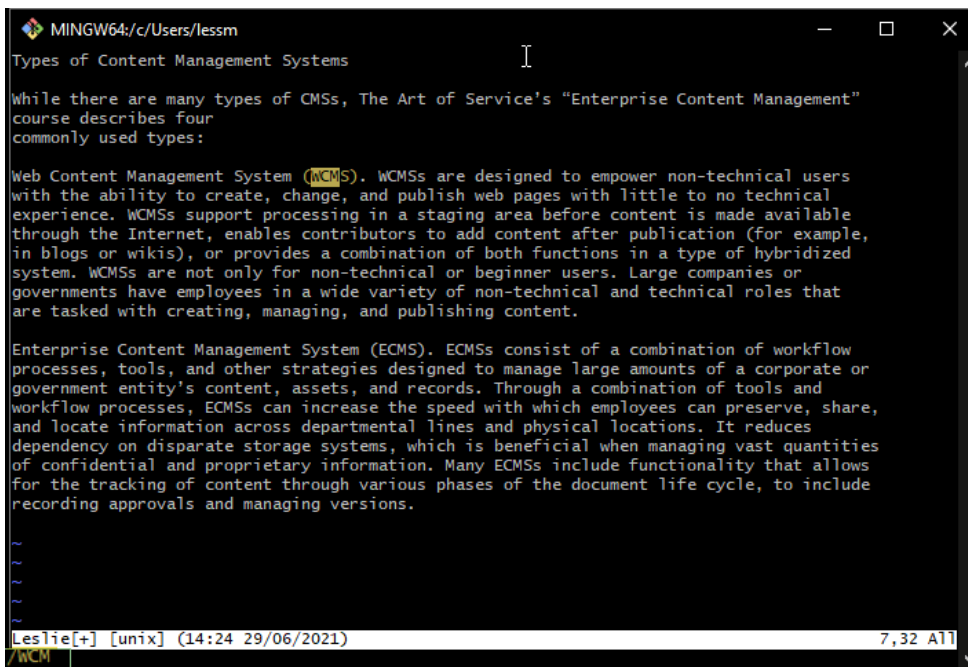


Figure 8. This example of a simple search on “WCMS” shows the search text on the command line under the status bar. Vim locates the first occurrence and highlights it in the text of the document as it is entered by the user.

3. Press **Enter** and the cursor moves to the word or phrase:

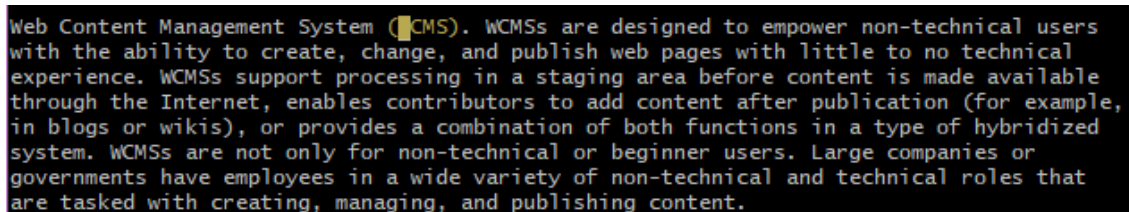


Figure 9. Depiction of the first occurrence of the searched word or phrase.

4. Press **n** to move the cursor to the next occurrence of the word:

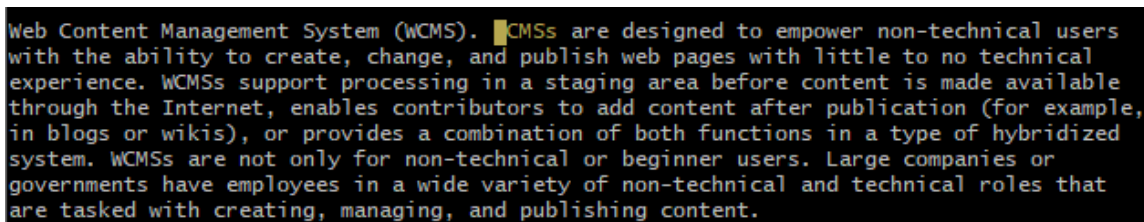


Figure 10. Depiction of the next occurrence of the searched word or phrase.

*Note:* Continue pressing **n** to search through the file as needed. Once Vim reaches the end of the file content, the message `Search hit BOTTOM, continuing at TOP` displays beneath the status bar.

- Press **N** to search in the opposite direction, as shown in Figure 11:

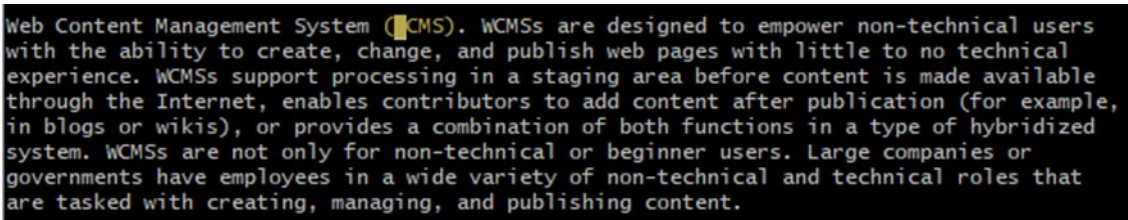


Figure 11. Demonstration of search in the opposite direction from that shown in Figure 7.

## Help

Vim has a well-defined help system. Contains a list of useful help commands, along with a description of the function for each:

Command	Function
:help	Opens the main online-help file in Vim.
Ctrl + ]	Opens a specific help topic file on which the user has placed the cursor prior to performing the command.
Ctrl + o	Goes back up one level in the help topics. Repeat this command to go back to a previous level, as necessary.
:q	Exits help.

Table 7. Help Commands [4]

Perform search commands in **normal mode** to avoid disabling the command and entering it as text in the document.

To access and view topics in Vim’s on-line help:

- Press the **Esc** key to ensure Vim is in normal mode.
- Type **:help** on the command line and press **Enter** to display the primary on-line help file, as shown in Figure 12:

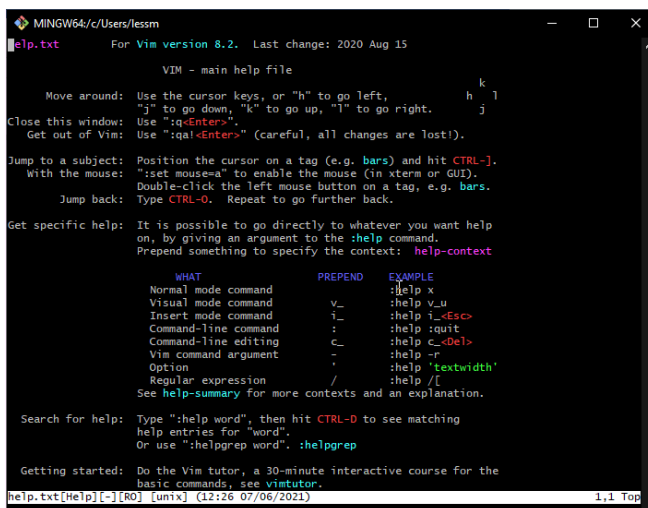
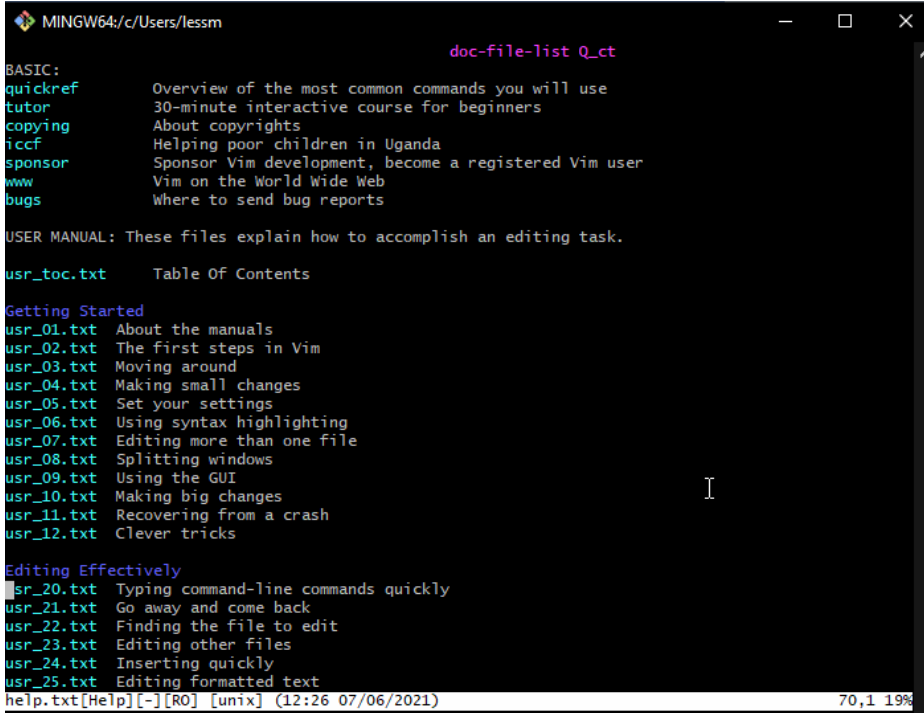


Figure 12. The help file contains instructions for using help at the top of the screen.



- Use cursor keys to move through the on-line help and view the list of available topics from the table of contents, as shown in Figure 13:



```
MINGW64:/c/Users/lessm
doc-file-list Q_ct
BASIC:
quickref  Overview of the most common commands you will use
tutor     30-minute interactive course for beginners
copying   About copyrights
iccf      Helping poor children in Uganda
sponsor   Sponsor Vim development, become a registered Vim user
www       Vim on the World Wide Web
bugs      Where to send bug reports

USER MANUAL: These files explain how to accomplish an editing task.

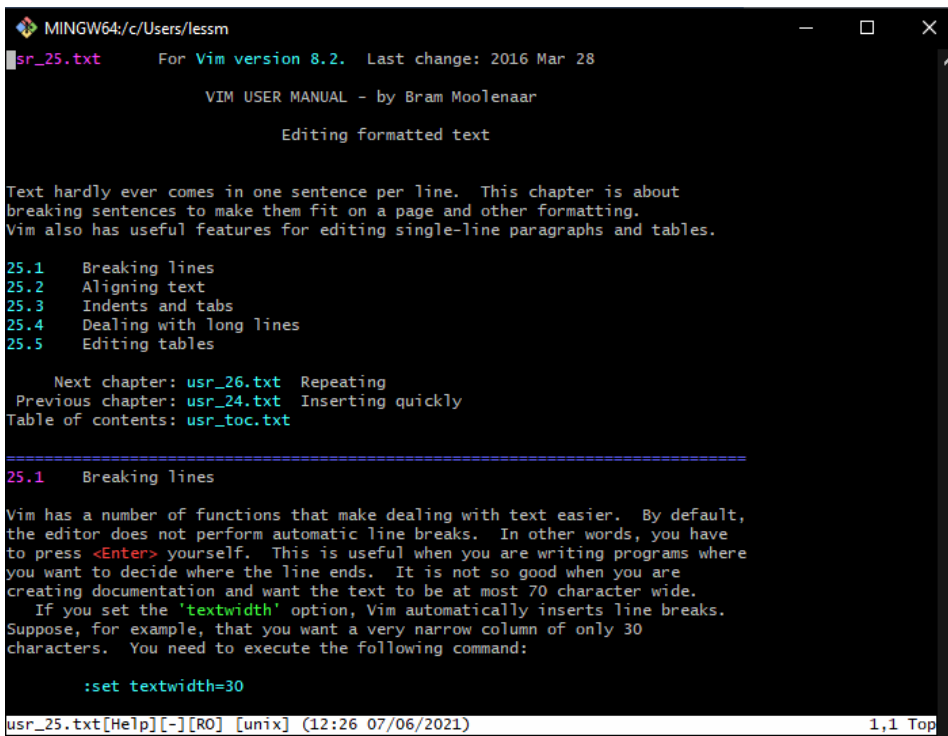
usr_toc.txt  Table Of Contents

Getting Started
usr_01.txt  About the manuals
usr_02.txt  The first steps in Vim
usr_03.txt  Moving around
usr_04.txt  Making small changes
usr_05.txt  Set your settings
usr_06.txt  Using syntax highlighting
usr_07.txt  Editing more than one file
usr_08.txt  Splitting windows
usr_09.txt  Using the GUI
usr_10.txt  Making big changes
usr_11.txt  Recovering from a crash
usr_12.txt  Clever tricks

Editing Effectively
usr_20.txt  Typing command-line commands quickly
usr_21.txt  Go away and come back
usr_22.txt  Finding the file to edit
usr_23.txt  Editing other files
usr_24.txt  Inserting quickly
usr_25.txt  Editing formatted text
help.txt[Help][~][R0] [unix] (12:26 07/06/2021) 70,1 19%
```

Figure 13. The Table of Contents for Vim’s on-line help.

- Place the cursor on a file name and press **Ctrl + J** to open the topic, as shown in Figure 14:



```
MINGW64:/c/Users/lessm
usr_25.txt  For Vim version 8.2. Last change: 2016 Mar 28
           VIM USER MANUAL - by Bram Moolenaar
           Editing formatted text

Text hardly ever comes in one sentence per line. This chapter is about
breaking sentences to make them fit on a page and other formatting.
Vim also has useful features for editing single-line paragraphs and tables.

25.1  Breaking lines
25.2  Aligning text
25.3  Indents and tabs
25.4  Dealing with long lines
25.5  Editing tables

Next chapter: usr_26.txt Repeating
Previous chapter: usr_24.txt Inserting quickly
Table of contents: usr_toc.txt

=====
25.1  Breaking lines

Vim has a number of functions that make dealing with text easier. By default,
the editor does not perform automatic line breaks. In other words, you have
to press <Enter> yourself. This is useful when you are writing programs where
you want to decide where the line ends. It is not so good when you are
creating documentation and want the text to be at most 70 character wide.
If you set the 'textwidth' option, Vim automatically inserts line breaks.
Suppose, for example, that you want a very narrow column of only 30
characters. You need to execute the following command:

:set textwidth=30

usr_25.txt[Help][~][R0] [unix] (12:26 07/06/2021) 1,1 Top
```

Figure 14. Help topics in Vim’s online help can contain additional files or headings through which users can “drill down” to learn more about a specific area of a topic.

- Repeat Step 4 if you want to access more specific information about a topic as shown in Figure 15:

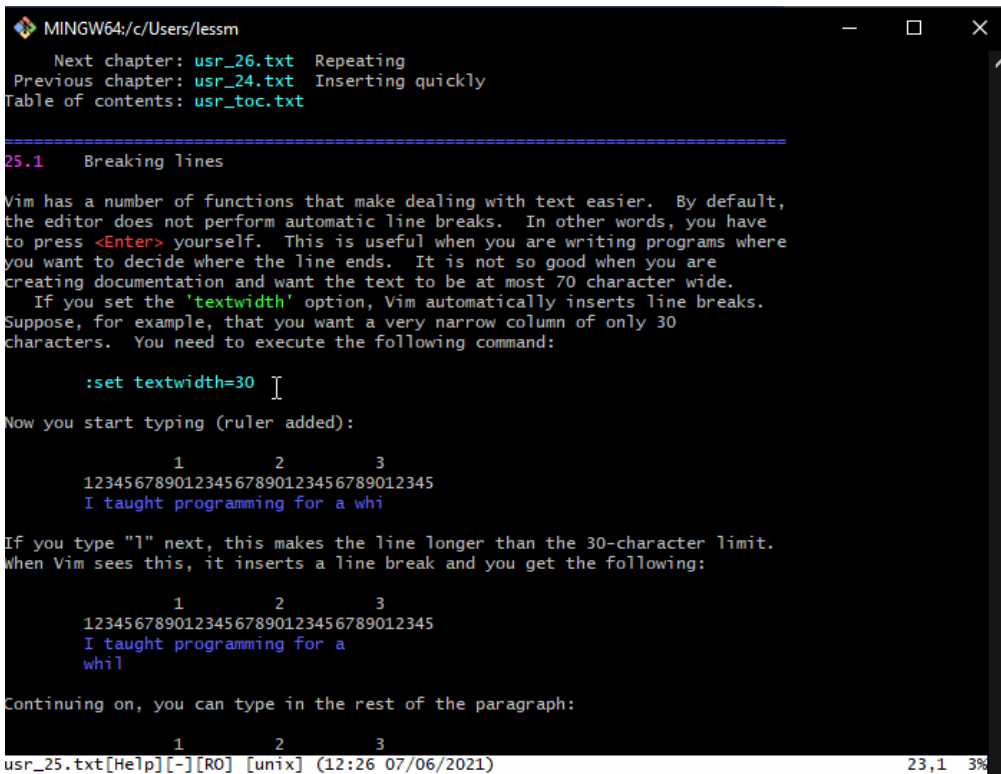


Figure 15. A help subtopic of the original topic depicted in Figure 14.

- Press **Ctrl + O** to back out of a topic and return to a previous screen in help. Repeat this step as necessary.
- Type **:q** to exit help. [4]

## Additional Resources

Vim is a robust and powerful text editor. This tutorial is intended to get you started in using the application, but to learn more about Vim, consider the following resources:

- [Vim.org](#) – Website dedicated to using and improving Vim
- [Vim Adventures](#) – Teaches Vim through an on-line game
- [Computer Hope](#) – Contains a Vim tutorial
- [Vim Tutorial](#) – Utube video by Ben Awad
- [Vim Cheat Sheet](#) – Lists and describes Vim commands

## References Cited

- [1] Computer Hope. "Linux vim command" March 13, 2021 [accessed June 28, 2021]  
<https://www.computerhope.com/unix/vim.htm>
- [2] Linux.org. "Linux Beginner Tutorials" April 25, 2017 [accessed June 28, 2021]  
<https://www.linux.org/threads/clear-clear-your-terminal-screen.78/>
- [3] Linuxize. "How to Save a File in Vim / Vi and Quit the Editor" October 2, 2020 [accessed June 29, 2021]  
<https://linuxize.com/post/how-to-save-file-in-vim-quit-editor/#open-a-file-in-vim--vi>
- [4] Pierce, Michael C and Ware, Robert K. Modified by Bram Moolenaar. "VIM Tutor, Version 1.7" N.d.  
[accessed June 28, 2021 – July 2, 2021]

This page is intentionally blank.