

Vim Survival Kit

Copyright 2020, email author for use permission.

Last Update: 2020-03-16

The Goal of this Kit

The **Vim Survival Kit** has been painstakingly crafted to be as short, and yet as complete, of a guide as possible to help you survive your first few edits (and beyond) in **Vim**!

In this session, we'll cover: - Creating, Opening, and Exiting a File - Opening an existing file - Making a Simple Change to a File - Changing a System File - Simple Navigation - Inserting, Copying and Deleting - Undoing and Redoing - Saving and/or Exiting - Resources for Getting Help

While hopefully very helpful, this guide is necessarily brief, and the Author urges you to complete the entire **Editing Text with Vim** course, there's so much more to discover about this great Text Editor!

Please Enjoy!

Ross Brunson

Training Architect II

ross.brunson@linuxacademy.com

Creating, Opening, and Exiting a File

Create and Save a New File with Vim

1. Open Vim `vim`
2. Press the `i` key and note the bottom left says you are in Insert Mode, and type in some text such as:
Mimsy were the borogroves or It was a dark and stormy night
3. When done entering the text, use `ESC` to return to Command Mode
4. Write the new file to disk with: `:w myfirstnovel.txt`
5. Then exit Vim with:
`:q` **Note:** You can specify a filename at the outset, such as in Step 1, and then saving and exiting can skip Step 4, and in Step 5 the command would then be:

```
:wq
```

Opening an existing file, exiting without changes

1. Edit the `/etc/hosts` file with:

```
vim /etc/hosts
```

2. Press the **i** key to go to Insert Mode, which will cause a `W10: Warning: Changing a readonly file` error message
3. Use **ESC** to return to Command Mode and **u** to ensure that no changes have been made, indicated by a message at the status line of `Already at oldest change`
4. Exit the file with:

```
:q!
```

Making a Simple Change to a File

Note: The `vim80` portion of the command line may be `vim81`, or even `vim82`, use the **TAB** key to find what is there.

1. Copy a **Vim** documentation file for our use:

```
cp /usr/share/vim/vim80/doc/help.txt ~/vimhelp.txt
```

2. Edit the file with:

```
vim ~/vimhelp.txt
```

3. Navigate to the text **VIM - main help file**
4. Change the word **VIM** to **Vim** by putting the cursor on the **I** and pressing **cw** to change the rest of the word to **im**
5. Then hit **ESC** to exit Insert Mode
6. Navigate to the **m** in **main** and use the **~** character to change it to **M**
7. Then do the same for the **h** in **help** and the **f** in **file**
8. Press **ESC** to ensure you are back in Command Mode
9. Save and exit with:

```
:wq
```

Changing a System File

1. Edit the `/etc/hosts` file with:

```
sudo -i vim /etc/hosts
```

2. Supply the `cloud_user`'s password (or the required user's)
3. Using the cursor keys (if necessary), go to the line that reads:

```
127.0.0.1      localhost
```

5. Press `A` which enters Insert Mode, and moves you to the end of the line
6. Now add a space and `snowblower` to the end of the line so that it reads:

```
127.0.0.1      localhost snowblower
```

7. Save and exit the file by first pressing `ESC` and then `ZZ`
8. Verify the change was written with:

```
grep snowblower /etc/hosts
```

9. Confirm that the name is resolvable with:

```
ping -c 4 snowblower
```

10. The ping should work properly, if not, check the line to ensure there is a space between `localhost` and `snowblower`.

Simple Navigation

1. Edit the `~/vimhelp.txt` file

```
vim ~/vimhelp.txt
```

2. Go to the top of the file with `gg`
3. Go to the bottom of the file with `G`
4. Go to line 25 with `25G`
5. Go to 50% of the way through the file with `50%`
6. Go to the top with `gg`
7. Press `w` 5 times to move forward 5, then `gg` to return to the top of the file
8. Press `25w` to move forward 25 words, then `5b` to move backwards 5 words
9. Use `h,j,k,l` to move around, then use the cursor keys
10. Press `gg` to go to the top
11. Use `/Vim`, `ENTER` to find the first instance of `Vim`
12. Use `n` to go to the next and subsequent instances
13. Use `N` to go back to previous instances
14. Go to the bottom of the file using `G`
15. Use `?help` to find backwards
16. Exit without saving using `q!`

Inserting, Copying and Deleting Text

1. Edit the `~/vimhelp.txt` file

```
vim ~/vimhelp.txt
```

2. Go to the top of the file with `gg` and then navigate to the line that starts with

```
Get out of Vim
```

3. Press `o` to add a line under the current line
4. On that line, add the below text, *taking care to line up the `:` with the other instances of `:`* on other lines

```
ATTENTION: Do NOT Reboot to Exit Vim!
```

6. When added, hit the `ESC` key to return to Command Mode, then use `^` to go back to the front of the word `ATTENTION`, then `0` (zero) to go to the very beginning of the line
7. Then place the cursor on the empty line between your current line and the one that reads `Jump to a subject` and delete the empty line by pressing `dd`
8. Now copy the `ATTENTION` line by moving to it and pressing `yy`
9. Go to line that starts with `Jump Back` and use `p` to paste the copied line below it
10. Then place your cursor on the blank line in between the current line and the one that begins with `Get specific help` and press the `yy` keys to copy the blank line
11. Then create 4 more blank lines by using `4p`
12. Now highlight or select all of the blank lines but the last one before `Get specific help` by putting your cursor on the top blank line, pressing `V` and moving your cursor to the next-to-last blank line (selecting 4 lines) and then delete those lines by pressing `d` once.
13. While on the remaining blank line below the copied line beginning with `ATTENTION:`, press `i` to enter Insert Mode and use `Space` to insert spaces until the cursor is under the `D` in `Do NOT` and add the text `You'll Regret it!` and when done press `ESC` to return to Command Mode
14. Move the cursor to the line above the current one, which puts your cursor in between the words `to` and `Exit` on that line and press `J` to join the line below to the end of the current line.

Undoing and Redoing

14. Now undo your changes one at a time by pressing `u`. You'll tire quickly of this, so hold down the `u` key until all changes to the file are reversed.
15. When you have reached the last undo possible, you will see the status line reflect with the text `Already at oldest change`.
16. Restore all changes to the file by pressing `Ctrl-r` repeatedly, noting what each "change" is made up of, and also what the status line says about how many changes were made.
17. When all changes have been redone, you'll see `Already at newest change`.

Note: For the next task to work properly, ensure that you *redo* all the changes.

Saving and/or Exiting

1. Save your changed buffer to a new file with by typing:

```
:w ~/changedhelp.txt
```

2. Look to see if you are in that buffer, or in the old buffer by pressing:

```
Ctrl-g
```

3. Confirm that there is only a single buffer, and that it's the vimhelp.txt buffer including changes with:

```
:ls
```

5. Exit the changed buffer without saving any changes with:

```
:q!
```

Resources for Getting Help

1. Run `vim` with no file argument:

```
vim
```

2. In Command Mode, use `:help` to get into Help Mode
3. Exit Help Mode with `:q` just to show **Vim** you can!
4. Invoke help for how to move around with `:h motion`
5. Peruse the text, then press the `Ctrl-f` key combo to move a screen forward, then use `Ctrl-b` to move back a screen
6. While in the `motion` help page, bring up the help for `Ctrl-f` with `:h Ctrl-f` and read about that
7. Look up the help for how to load Vim with just it's defaults, so when something goes wrong you can troubleshoot it with `:h --clean`
8. When done, quit out of Help Mode with `:q` and completely out of Vim with by pressing `ESC` to ensure you're in Command Mode, and then `:q!` to completely exit.
9. Navigate to the Vim help documentation directory with:

```
cd /usr/share/vim/vim80/doc (May be vim81!)
```

10. Use the following command to find what files in the directory have the "motion" keyword in them:

```
grep -wn motion *.txt
```

11. Note the results. The file **visual.txt** should be the last one listed.

12. Edit the **visual.txt** file and go straight to line 295 (or the line number indicated) where **motion** is located with:

```
vim +295 visual.txt
```

13. Note that you are placed on the beginning of line 295.
14. Exit the **visual.txt** file with:

```
:q!
```

16. Rerun the command to edit **visual.txt** with a search for motion instead:

```
vim +/motion visual.txt
```

Note: You may have a message about it being readonly, hit `ENTER` to continue

17. When the file is loaded, set search highlighting on with `:set hlsearch` and you'll see that **motion** is highlighted. You can ignore it if there are other highlighted words than **motion**, those are not actual highlights, just color codes.
18. Exit Vim with `:q!` and rerun the previous grep command, adding `| less` to the end of it so that the output will go into the `less` pager and you can scroll through and see what other places your search term is found!

```
grep -wn motion *.txt | less
```

19. Repeat this anytime you need to find a term in the Vim help files, it's a marvelous source of good information for the curious **Vim-ster**!

Congratulations, you have made it through the Vim Survival Kit tasks!