Vimrc Configuration Guide - How to Customize Your Vim Code Editor with Mappings, Vimscript, Status Line, and More

Configuring your .vimrc file lets you use the full power of Vim. With a customized .vimrc file you can increase your Vim powers tenfold.

In this article I will show you a few ways you can customize your .vimrc file.

I will go over:

Basic Settings

Plugins

Folding

Vimscript

Status line

First create the following directory structure in your home directory.

Create a .vimrc file in your home directory.

How to Update Basic Settings in Vim

First let's add some basic settings that will improve your editing experience. I use double quote characters to comment out lines.

Add the following lines to your .vimrc file:

Syntax highlighting is very useful. The next line we add will enable syntax highlighting and make your code easier to read.

This is what it looks like before:

And after:

You can also choose to display line numbers to make navigating the file easier.

You can pinpoint exactly where the cursor is located by highlighting the line it is on horizontally and vertically.

Add these lines to enable this feature.

Here are some more common setting that enhance the editing experience.  
Each line contains a comment above it explaining what it does.

Add the following lines to the .vimrc file.

Bash completion is a great feature which saves keystrokes by auto completing what you type. Vim has a similar feature called wildmenu.

Add the following lines to enable the wildmenu feature. You will see a list of files matching the file you are searching for. You can also enable auto completion to Vim.

Type for more information on specific commands.:help <command>

Example:

How to Fold Long Files in Vim

The .vimrc file can get long so organizing it into sections is a smart idea.  
Vim will allow you to fold long files to hide sections of text.

Add the following lines to the bottom of your .vimrc to organize the file into sections.

Save the .vimrc file with and source the .vimrc file like this to make the changes take effect. Now, once you move your cursor on a fold you can press::w:source ~/.vimrc

zo to open a single fold under the cursor.

zc to close the fold under the cursor.

zR to open all folds.

zM to close all folds.

Type for more information.:help folding

How to Add Plugins to Vim

You can add plugins to Vim to add extra functionality. Most people use a plugin manager to make plugin installation easy.

There are a variety of plugin managers we can use. I will show you how to install and use the vim-plug plugin manager.

To install the vim-plug plugin, run this command:

On Linux or Mac OS.

On Windows with Powershell.

Add the and lines in the plugins section. The plugins we install will be added between the two function calls.call plug#begin('~/.vim/plugged')call plug#end()

Save the .vimrc file with the command and source the .vimrc file with this command to make the changes take effect.:w:source ~/.vimrc

Now type to download and install the two plugins.:PlugInstall

How to Map Keyboard Shortcuts in Vim

In the mapping section we will add shortcuts to making typing longer commands easier. This will save you key strokes and lots of time, especially for long commands.

Key mapping syntax is like this:

map\_mode <what\_you\_type> <what\_is\_executed>

Popular Mapping Modes in Vim

Here are a few popular mapping modes and probably the most useful and important.

nnoremap – Allows you to map keys in normal mode.

inoremap – Allows you to map keys in insert mode.

vnoremap – Allows you to map keys in visual mode.

A common mapping example is to map 'jj' to the escape key. You will be pressing the escape key a lot. The escape key is in the far corner of the keyboard.  
The letter 'j' is in the middle of the keyboard so it is easier to press 'jj' instead of reaching for the escape key.

This is how you would map the escape key to .jj

inoremap jj <esc>

How to Use Mapleader in Vim

Mapleader will allow you set a key unused by Vim as the key.  
The leader key, in conjunction with another key, will allow you to create new shortcuts. <leader>

The backslash key is the default leader key but some people change it to a comma .","

let mapleader = "\"

With the leader key mapped to backslash, I can use it like this:

Turn off search highlighting by pressing .\\  
nnoremap <leader>\ :nohlsearch<CR>

Here are some common mappings that people use. See the comments above each line for the explanation.

Add this code in the mappings section:

Type for more information.help: map-modes

How to Add Some Vimscripting

Vimscript is a scripting language that lets you create scripts using variables, if else statements, and functions. Auto commands are waiting for events to occur in order to trigger a command.

Read Learn Vimscript the Hard Way for more information on Vimscript.

Type for more information on auto commands.:help autocmd

How to Add Color Schemes to Vim

You can easily add color schemes to Vim to change the default colors. Do a search for Vim color schemes and you will find many, many choices.

Installing a color scheme is a simple as adding a file to the directory.<colorscheme>.vim~/.vim/colors/

I will add the popular color scheme molokai:

To set the color scheme, type this command:

Example color schemes:

color schemes: molokai, base16-tomorrow, blue, one

How to Configure the Status Bar in Vim

You can configure your Vim status bar with useful information. For example, configure the file type, total number of lines in the file, path to the file, column number, row number, percentage through file, and much more.

Add this code in the status line section:

%F – Display the full path of the current file.

%M – Modified flag shows if file is unsaved.

%Y – Type of file in the buffer.

%R – Displays the read-only flag.

%b – Shows the ASCII/Unicode character under cursor.

0x%B – Shows the hexadecimal character under cursor.

%l – Display the row number.

%c – Display the column number.

%p%% – Show the cursor percentage from the top of the file.

Type for more information.help: statusline

This is the complete .vimrc file.

Conclusion

In this article, I have only scratched the surface of how you can customize Vim.  
There are thousands of ways to configure and customize a .vimrc to your liking.  
You can even write your own plugins and color schemes and share them with the world.

I hope that you have learned a new trick or two by reading this article. So if you use Vim, don't leave home without a .vimrc file!

By the way, I used Vim to write this article.

Follow me on Github | DEV.to