How to add features to Vim editor

Vim is an editor that you can open to edit a configuration file that cannot be removed. On the other hand, if you regularly use Vim, you will know how powerful its editing features are.

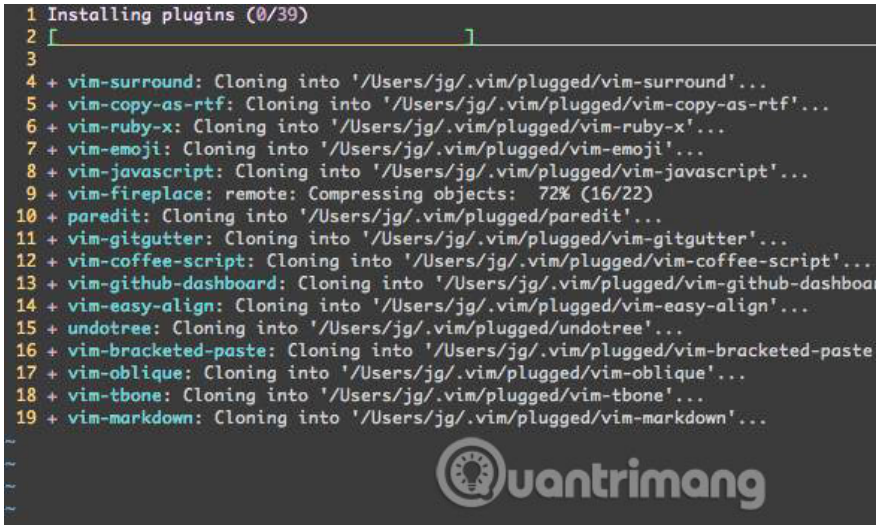
You probably already know about Vim. Vim is an editor that you can open to edit a configuration file that cannot be removed. On the other hand, if you regularly use Vim, you will know how powerful its editing features are. If running Linux or any other Unix distribution, it's worth learning about Vim.

By default, Vim lacks many of the features that users often find in modern text editors. Please install some Vim packages and variables into a powerful tool like Visual Studio Code, Sublime Text, etc.

Optimize Vim with features available in modern editors

1. Plugin management: Vim-Plug
2. Error checking: Syntastic
3. Complete code: YouCompleteMe
4. Search file: CtrlP
5. Browse file: NERDTree
6. Integrated Git: fugitive.vim

Plugin management: Vim-Plug



Another major feature in modern text editors is the ability to expand with plugins. Although Vim has added the original package management feature in version 8.0, many people find it quite 'cumbersome' compared to third-party package managers. One of the most popular package managers is Vim-Plug.

Before you can start using Vim-Plug, you need to install it. On Unix systems like Linux or macOS, run the following command in the terminal to download and install Vim-plug.

```
curl -fLo ~/.vim/autoload/plug.vim --create-dirs https://raw.githubusercontent.co
```

If you use Vim in Windows, you can install Vim-Plug by pasting the following command into PowerShell.

```
md ~vimfilesautoload $uri = 'https://raw.githubusercontent.com/junegunn/vim-plug/
```

Now, you will be able to install the plugin by adding them to the **~ / .vimrc** file **.** You need to add 2 new lines to the file:

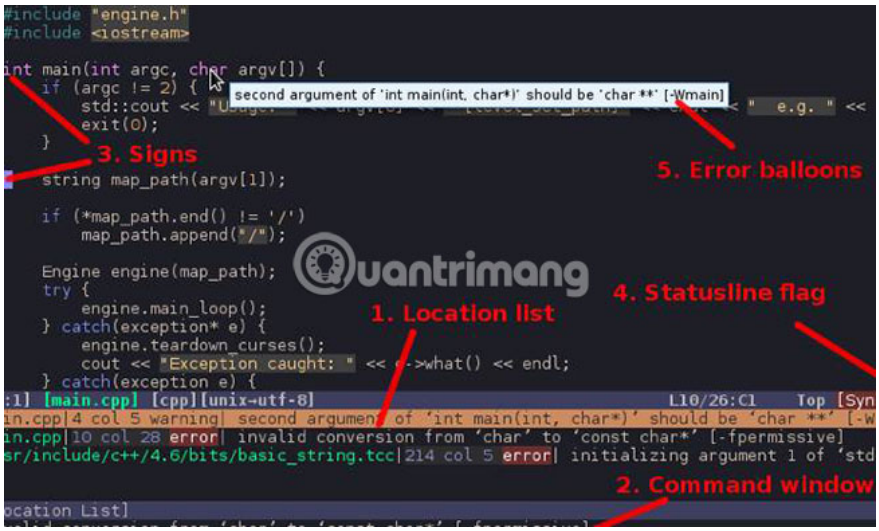
```
call plug#begin('~/.vim/plugged') call plug#end()
```

To install a plugin, add the **Plug,** plus a portion of the GitHub URL followed **http://www.github.com** in parentheses. For example, to install the Solarized palette, the configuration file will contain the following command:

```
call plug#begin('~/.vim/plugged') Plug 'altercation/vim-colors-solarized' call pl
```

For more information on how to install the package manager, see the Vim-Plug GitHub page (link: *https://github.com/junegunn/vim-plug)*

Error checking: Syntastic



Another feature many people need to rely on is notification when you write invalid code. This feature is commonly known as 'linting'. It doesn't help you not write code that can't run, but will show you basic syntax errors that you might not notice.

As its name suggests, Syntastic is a syntax checker plugin for Vim. Syntastic does not support all programming languages. Instead, you will need to install the syntax checker for the selected language (s). Syntastic will then integrate the checker into Vim, check the code every time you save the file.

Syntastic supports many popular languages, so it is possible that the language you use is also supported. For instructions on how to configure this plugin, see the GitHub Syntastic page (reference link: *https://github.com/vim-syntastic/syntastic).*

Complete code: YouCompleteMe

Syntax checking is a very useful feature, but if you have used Visual Studio Code or an editor with similar features, you may find something missing. That is the code completion feature, also called Intellisense in Visual Studio. If you use Vim for more than editing configuration files, the code completion feature will make your life a lot easier.

The code completion feature makes writing code easier, by making suggestions as you type. This is great if you use a method that includes many nested functions. At that point, you don't need to remember the whole string anymore.

YouCompleteMe is a code completion tool for Vim and one of the powerful plugins you can install. YouCompleteMe is also somewhat harder to install than other plugins. You can install basic things with package managers like Vim-Plug, but you will need to compile it.

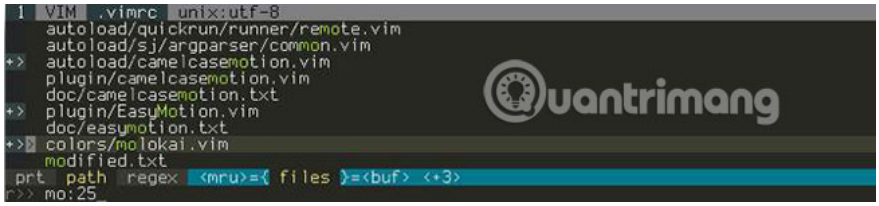
The easiest way to compile the plugin is to use the included install.txt script. To do this on macOS or Linux, enter the following:

```
cd ~/.vim/bundle/YouCompleteMe ./install.py --clang-completer
```

Note that on Linux, you will have to install development tools, CMake and necessary headers before you can compile YouCompleteMe.

For instructions on how to install and compile YouCompleteMe on other systems, see the YouCompleteMe GitHub page (reference link: *https://github.com/Valloric/YouCompleteMe).*

Search file: CtrlP

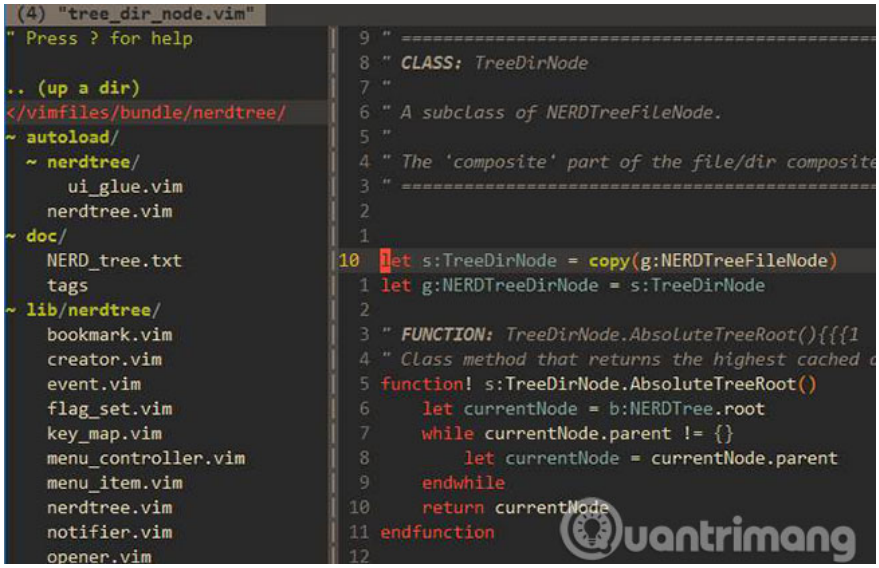


If you work on a project with many different files, Vim's file opening method can disappoint you. The command **: e** has the basic auto-completion feature, but you still need to know where your file is located. You can use the command line to find the file, but it's better if you can do this right in Vim.

Fortunately, the CtrlP plugin can help search files and more. CtrlP page GitHub description CtrlP is a tool to find file buffer, mru, tag, etc. for Vim. This plug-in is similar to Sublime Text's **'Goto Anything'** command, with a shortcut of **Ctrl + P** or **Cmd + P.**

This feature or an equivalent feature can be found in most modern text editors and if you find yourself missing, it's great to have it in Vim.

Browse file: NERDTree



You may prefer a more traditional file browser. If you remember the screen showing the left panel of the files found in many editors, you'll be happy to know it's available in Vim, thanks to the NERDTree plugin.

Unlike the left menu in Sublime Text, Visual Studio Code and other editors, NERDTree is a comprehensive system explorer file. Instead of just displaying the project folder, you can navigate to any location on the

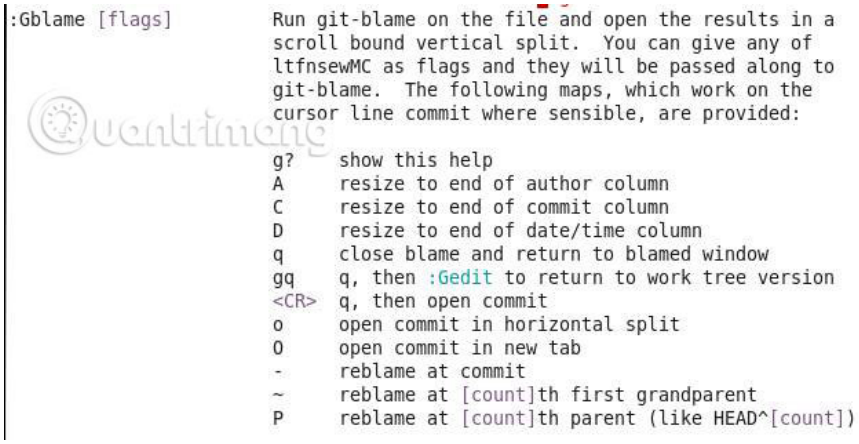
computer. If you work with files on multiple projects, this can be a very handy feature.

To open NERDTree inside Vim, simply use the command **: NERDTree.** If you don't like using the command, you can do this with the **~ / .vimrc option** as follows:

```
map :NERDTreeToggle
```

This will allow you to simply press **Ctrl + N** to open and close the NERDTree table.

Integrated Git: fugitive.vim



Git integration has become a must-have feature in modern text editors, so it's good to know that it's also available in Vim.

Run **: GStatus** will display the same result as when using the git status command. If you have completed your work on a file, run **: GCommit** %. This will allow you to edit the confirmation message inside the currently running Vim window.

There are too many commands to list here, but you can run any standard Git command by running: Git. For more information, see the fititive.vim page GitHub (reference link: *https://github.com/tpope/vim-fugitive).*

The tips above will help 'modernize' Vim, but they are not the only things you can do to customize the editor to your liking.

Hope you are succesful.