How to Install the `typst.vim` Plugin on Mint 22.1

By `kalwisti`

1. Introduction

This brief tutorial explains how to install a Vim plugin for Typst (`typst.vim`) which was created by `kaarmu` (Kaj Munhoz Arfvidsson). If you are familiar with Vim, this will give you an alternative to using the VS Code editor with your local Typst installation.

Plugins extend Vim's core functionality. As a user with limited Vim experience, the mechanics of installing a plugin were confusing to me. Fortunately, I found two helpful videos by Jay LaCroix and Eric Murphy which cover the basics of configuring Vim and installing plugins. I recommend that you begin by watching these tutorials, as they will make it easier to follow the instructions.

The method below is a combination of some tips offered by Jay and Eric, as well as the instructions presented by Junegunn Choi on GitHub. It is straightforward and effective.

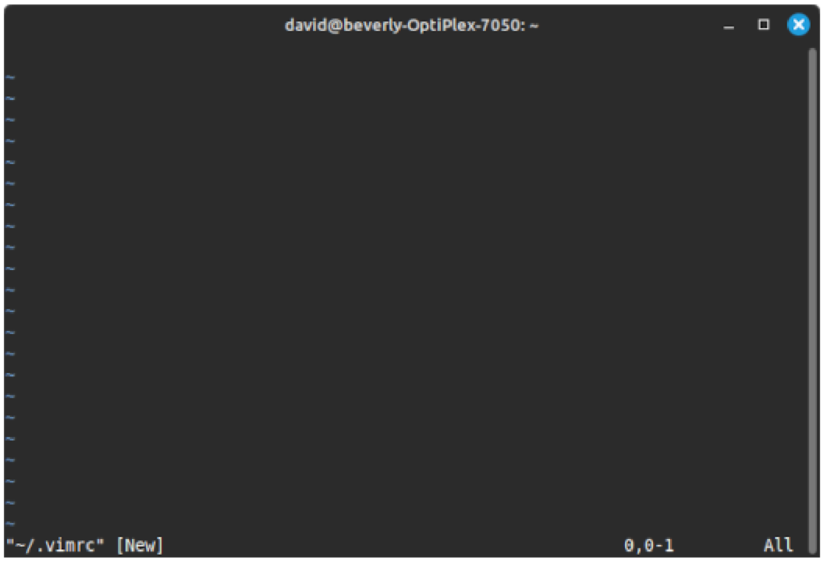
2. Prerequisites

Check that you already have these packages installed on your system:

- The Typst pre-built binary
  - My previous post explains how to install the Typst compiler locally.
- Vim
- `curl`
- `git`

3. Create Your Local `.vimrc` File

When you start a new file in Vim, the window is almost blank. The lines are unnumbered and the blue tildes show you where a line *is not*, i.e., that there is nothing there—yet.



```
"~/.vimrc" [New]                                   0,0-1         All
```

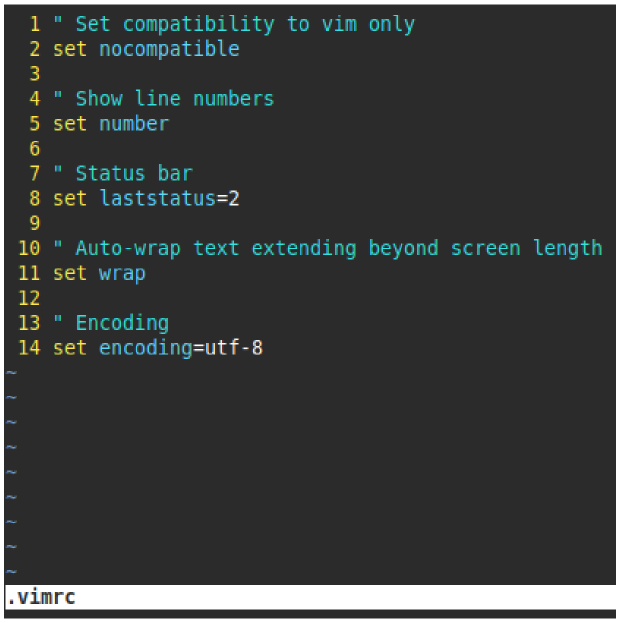
We will begin by creating a local `.vimrc` file and adding some simple customizations. (This file contains Vim's core global settings.)

- Open a Terminal/Konsole and type:

```
$ vim ~/.vimrc
```

- Go into Insert mode and add the following:

```
" Set compatibility to vim only
set nocompatible

" Show line numbers
set number

" Status bar
set laststatus=2

" Auto-wrap text extending beyond screen length
set wrap

" Encoding
set encoding=utf-8
```



```
 1 " Set compatibility to vim only
 2 set nocompatible
 3
 4 " Show line numbers
 5 set number
 6
 7 " Status bar
 8 set laststatus=2
 9
10 " Auto-wrap text extending beyond screen length
11 set wrap
12
13 " Encoding
14 set encoding=utf-8
~
~
~
~
~
~
~
~
~
.vimrc
```

Note:

The double quote (") at the start of lines 1, 4, 7, 10 and 13 is part of Vim's comment syntax. It causes the whole line to be ignored.

You do not need a closing quote.

- Save your changes and exit Vim.

4. Download `vim-plug` and Place It in `autoload` Directory

The next step is to download the `vim-plug` Plugin Manager from Junegunn's GitHub repository and place it in the `autoload` directory.

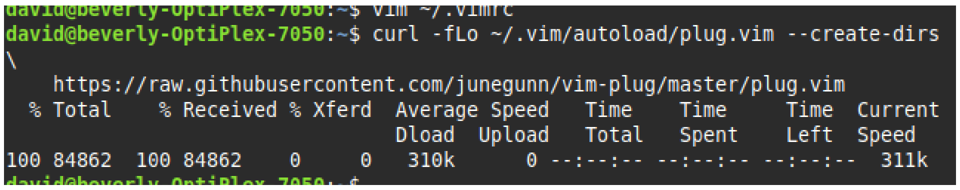
Note:

There are multiple methods to install plugins in Vim. This how-to uses the easy, dependable and performant "vim-plug" Plugin Manager.

- Open a Terminal/Konsole and type:

```
curl -fLo ~/.vim/autoload/plug.vim --create-dirs \
    https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
```

Afterwards you will see something like the screen below:



```
david@beverly-OptiPlex-7050:~$ vim ~/.vimrc
david@beverly-OptiPlex-7050:~$ curl -fLo ~/.vim/autoload/plug.vim --create-dirs
\
    https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 84862  100 84862    0     0   310k      0 --:--:-- --:--:-- --:--:--  311k
david@beverly-OptiPlex-7050:~$
```

5. Add `vim-plug` Section to Your `.vimrc`

- Open a Terminal/Konsole and type:

```
$ vim ~/.vimrc
```

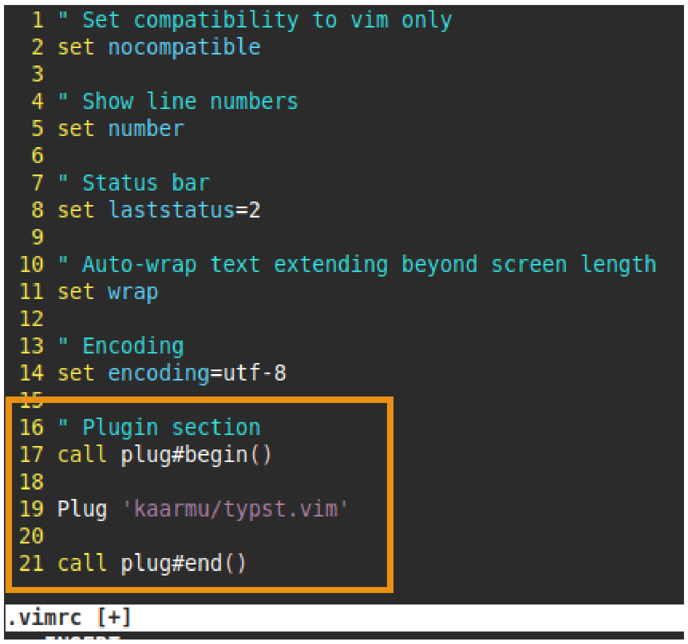
- At the end of the file, add:

```
" Plugin section
call plug#begin()

Plug 'kaarmu/typst.vim'

call plug#end()
```

Note:

This section • begins with the command `call plug#begin()`, • lists the plugins with a `Plug` command, and • ends with the `call plug#end()` command.

The `'kaarmu/typst.vim'` entry designates the actual `typst.vim` plugin. It follows the pattern `(developer's) username` + `repository name`.



```
" Set compatibility to vim only
set nocompatible

" Show line numbers
set number

" Status bar
set laststatus=2

" Auto-wrap text extending beyond screen length
set wrap

" Encoding
set encoding=utf-8

" Plugin section
call plug#begin()

Plug 'kaarmu/typst.vim'

call plug#end()
.vimrc [+]
-- INSERT --
```

- Save your changes and exit Vim.

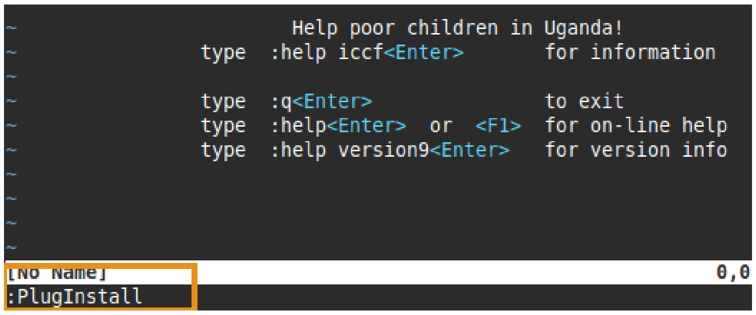
Note:

You can add more plugins later, if you wish. Additional plugin entries should be placed between the `plug#begin` and `plug#end` lines. The plugin entries should follow the naming convention described above.

5.1. Install the `typst.vim` Plugin

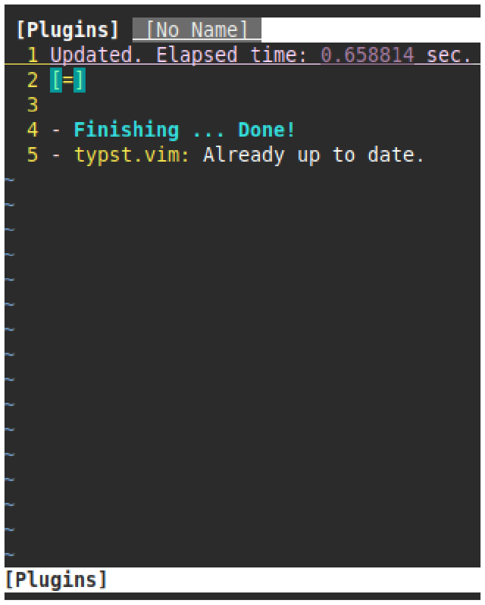
- Restart Vim: `$ vim`
- In Command mode, type: `:PlugInstall`

(This opens the plugin manager within Vim and proceeds to install all plugins listed in the `.vimrc` file. Installed plugins automatically load the next time Vim is started.)



```
~                       Help poor children in Uganda!
~               type  :help iccf<Enter>       for information
~
~               type  :q<Enter>               to exit
~               type  :help<Enter>  or  <F1>  for on-line help
~               type  :help version9<Enter>   for version info
~
~
~
~
[No Name]                                                   0,0
:PlugInstall
```

If this succeeds, you should see something like the screenshot below:



```
[Plugins]  [No Name]
  1 Updated. Elapsed time: 0.658814 sec.
  2 [=]
  3
  4 - Finishing ... Done!
  5 - typst.vim: Already up to date.
~
[Plugins]
```

5.2. Other Plugin Commands

You might find these three commands useful; they are also issued from Vim's Command mode:

`:PlugUpdate` to install or update the plugins

`:PlugDiff` to review the changes from the last update

`:PlugClean` to remove plugins no longer in the list

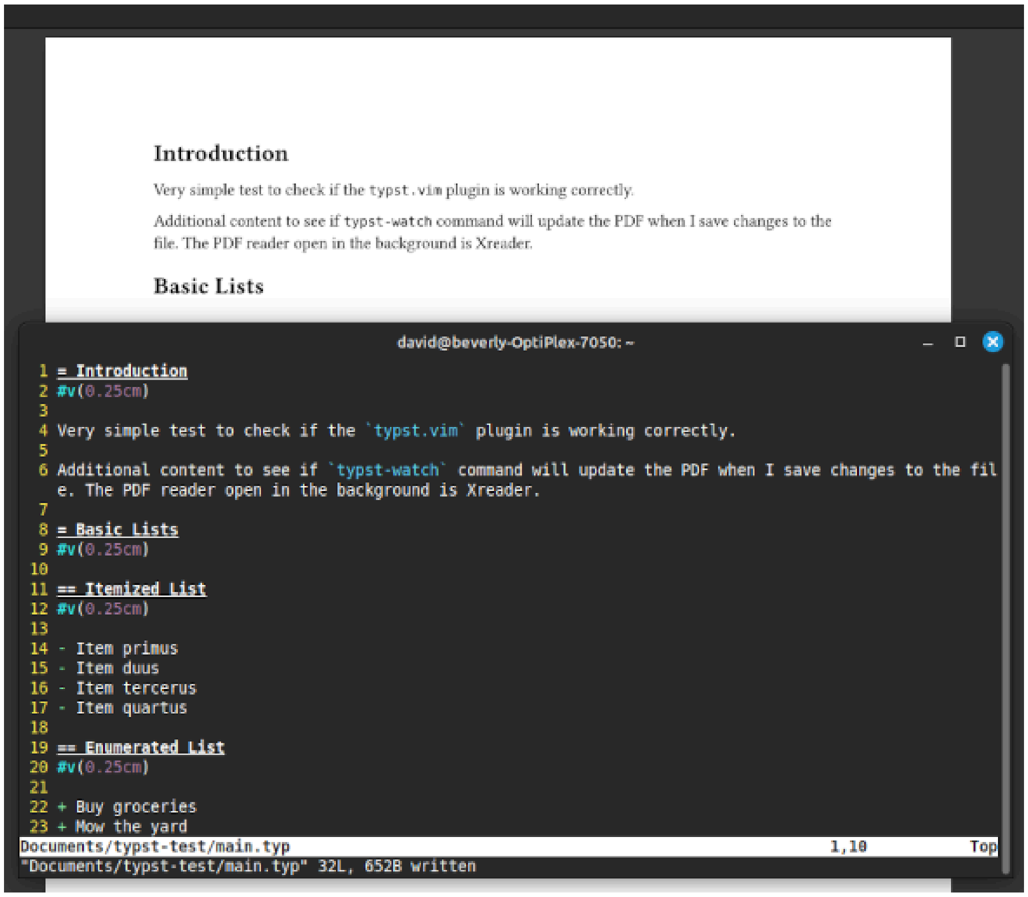
6. Basic Usage of `typst.vim` Plugin

6.1. `:TypstWatch`

- Use Vim to open your local Typst file and begin editing.
- In Command mode, type: `:TypstWatch`

The `typst-watch` command activates a mode where Typst continuously monitors the specified file for any changes. When you save your changes, Typst will recompile the file and update the displayed PDF.

- This command will open the compiled PDF in a PDF reader—Xreader, in the example below:



Introduction

Very simple test to check if the `typst.vim` plugin is working correctly.

Additional content to see if `typst-watch` command will update the PDF when I save changes to the file. The PDF reader open in the background is Xreader.

Basic Lists

```
= Introduction
#v(0.25cm)

Very simple test to check if the `typst.vim` plugin is working correctly.

Additional content to see if `typst-watch` command will update the PDF when I save changes to the file. The PDF reader open in the background is Xreader.

= Basic Lists
#v(0.25cm)

== Itemized List
#v(0.25cm)

- Item primus
- Item duus
- Item tercerus
- Item quartus

== Enumerated List
#v(0.25cm)

+ Buy groceries
+ Mow the yard
Documents/typst-test/main.typ                                1,10          Top
"Documents/typst-test/main.typ" 32L, 652B written
```

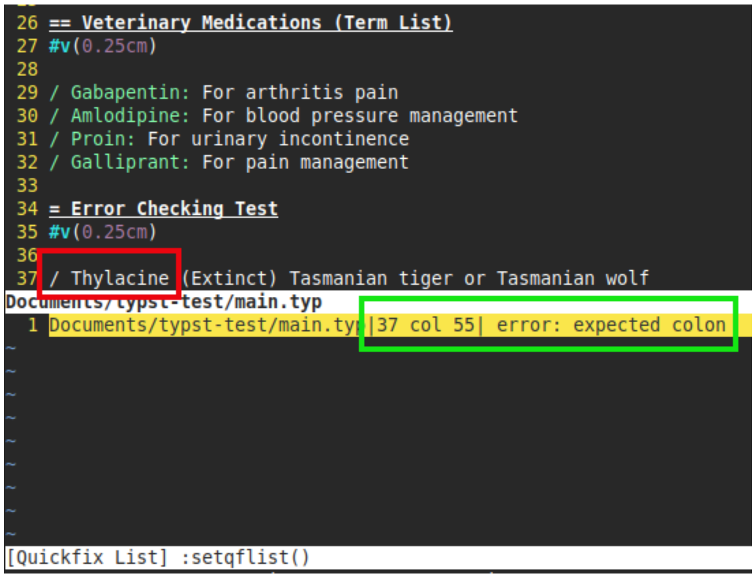
- You can leave Xreader open in the background. Typst will automatically update the PDF after you save changes to your source file.
- Although this setup is more like working with a traditional LaTeX editor (in the sense that it does not offer a live preview), it offers a writing environment with fewer distractions. Some people might find that appealing.

6.2. Error Checking

The `typst.vim` plugin includes error checking.

In the example below, I intentionally omitted the required colon following "Thylacine" in the term list. (It should be " / Thylacine: ")

When I saved my file, the plugin notified me that there was an error:



```
== Veterinary Medications (Term List)
#v(0.25cm)

/ Gabapentin: For arthritis pain
/ Amlodipine: For blood pressure management
/ Proin: For urinary incontinence
/ Galliprant: For pain management

= Error Checking Test
#v(0.25cm)

/ Thylacine (Extinct) Tasmanian tiger or Tasmanian wolf
Documents/typst-test/main.typ
  1 Documents/typst-test/main.typ|37 col 55| error: expected colon
[Quickfix List] :setqflist()
```

7. Additional Resources

Arfvidsson, Kaj Munhoz (`kaarmu`). "`typst.vim`." GitHub. https://github.com/kaarmu/typst.vim

Choi, Junegunn (`junegunn`). "`vim-plug`." GitHub. https://github.com/junegunn/vim-plug?tab=readme-ov-file#installation

LaCroix, Jay. "Intro to Vim Customization: Configuration and Plugins." *YouTube*, 30 Jun. 2021. (13 min., 34 sec.) https://www.youtube.com/watch?v=zE0hno3vV9M

Murphy, Eric. "How to Install and Manage Vim Plugins (The Easy Way)." *YouTube*, 11 Oct. 2021. (8 min., 52 sec.) https://www.youtube.com/watch?v=MztQMLyyCnU

Have fun using Typst with Vim as your editor!