

A *typst.vim* kiterjesztés telepítése

PCLinuxOS Magazine - 2025. augusztus

Írta: David Pardue (kalwisti)

Ez a rövid leírás elmagyarázza a Vim kiegészítő Typst-ba telepítését ([typst.vim](#)), amit kaarmu (Kaj Munhoz Arfvidsson) fejlesztő készített. Ha jártas vagy Vim-ben, akkor ez alternatívája lehet a helyi telepítésű Typst VSCodium (vagy VS Code) szerkesztőjének. Noha, ezzel nagyjából olyan lesz, mintha a hagyományos LaTeX szerkesztővel dolgoznánk – abban az értelemben, hogy nincs élő előnézet –, kevesebb fennakadással működő írási közeget biztosít.

A kiegészítő növeli a Vim alapképességeit. Korlátozott Vim-gyakorlással rendelkezőként a kiegészítő telepítése elsőre kicsit furcsa volt nekem. Szerencsére találtam két hasznos videót [Jay LaCroix](#)-tól és [Eric Murphy](#)-tól, ami a Vim beállításának alapjaival és kiegészítők telepítésével foglalkozik. Javasolom, hogy ezekkel az oktató anyagokkal kezd, könnyebben követhetővé teszik az utasításokat.

A következő „hibrid” eljárás Jay és Eric néhány ajánlott tippjének és [Junegunn Choi](#) GitHub-on található instrukcióinak kombinációja. Egyértelmű és hatékony.

Előfeltételek

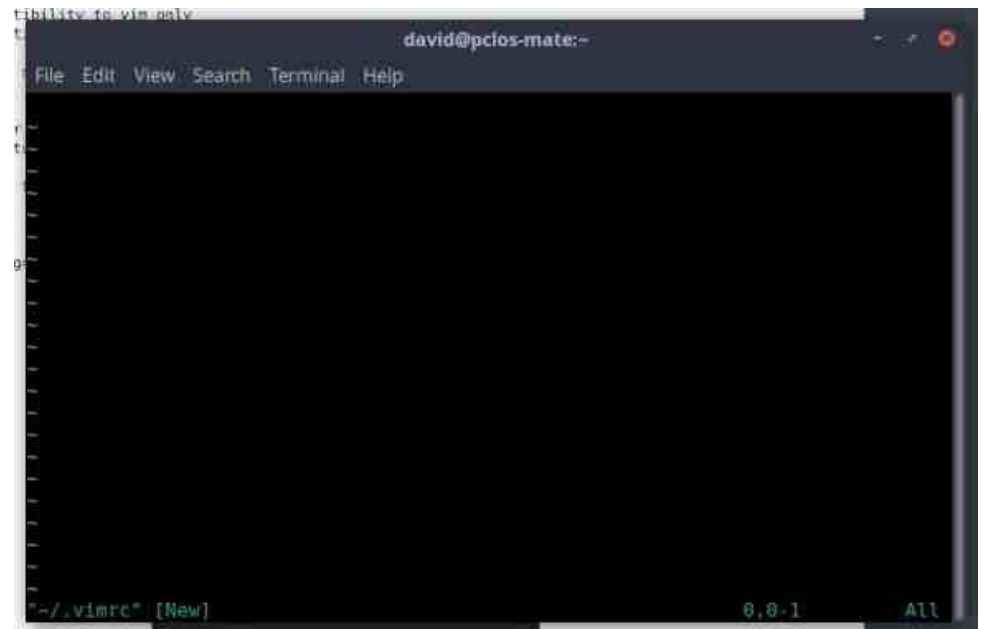
Ellenőrizd, hogy a következő csomagok telepítve vannak-e a rendszeredben:

- Typst [binárisa](#) telepítve a rendszeren. A [korábbi cikkem](#) leírja, hogyan telepítsük a Typst-fordítót a PCLinuxOS-re helyileg.
- Vim (csomagok: vim-minimal, vim-common, vim-enhanced és vim-X11) a PCLinuxOS tárolóiból.

- curl
- git

Készítsd el a saját .vimrc fájlodat

Amikor új fájlba kezdesz Vim-mel, az ablak szinte teljesen üres lesz. A sorok számozatlanok és kék hullámvonal jelöli az üres sorokat, azaz semmi sincs ott – egyelőre.



Helyi .vimrc fájl elkészítésével és kis, alapszintű testre szabással kezdjük. (Ez a fájl tartalmazza a Vim globális beállításait.)

Nyiss terminált (konzolt) és írd be: **\$ vim ~/.vimrc**

Lépj inzertmódba és add hozzá a következőket:

```
" 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
```

```
david@pclos-mate:~  
File Edit View Search Terminal Help  
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

Vedd észre a dupla idézőjeleket (") az 1., 4., 7., 10. és 13. sor előtt. Ez a Vim kommentelő szintaxisa. Általa az egész sort negligálja. Záró idézőjel nem kell.

Mentsd a változásaidat és lépj ki a Vim-ből.

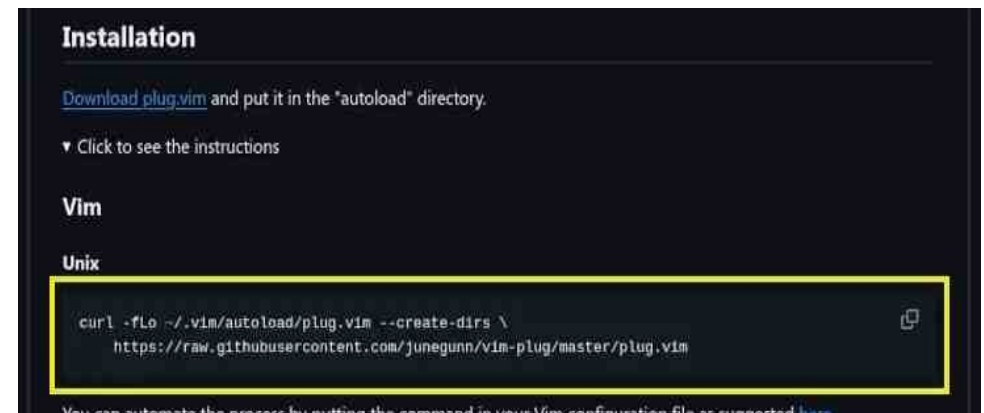
Töltsd le a vim-kiterjesztést és helyezd el az autoload könyvtárban

Következő a **vim-plug** kiterjesztéskezelő letöltése Junegunn GitHub tárolójából és elhelyezése az **autoload** könyvtárban.

Tudd, hogy több módon lehet a Vim-kiterjesztéseket telepíteni. Itt a könnyű, megbízható és működő „vim-plug” kiterjesztéskezelőt mutatom be.

```
Nyiss terminált (konzolt) és írd be (egyetlen sorba):  
curl -fLo ~/.vim/autoload/plug.vim --create-dirs  
https://raw.githubusercontent.com/junegunn/vim-plug/master/plug.vim
```

A fenti parancsot, ha akarod, közvetlenül is letöltheted Junegunn GitHub-os [instrukcióiból](#).



Ezután valami olyasmit látsz, mint ami következő képernyőképen van:

```
File Edit View Search Terminal Help
[david@pclos-mate ~]$ 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  443k    0 --:--:-- --:--:-- --:--:-- 445k
[david@pclos-mate ~]$
```

A .vimrc fájlodhoz add hozzá a vim.plug szekciót

Nyiss terminált (konzolt) és írd be `$ vim ~/.vimrc`

A fájl végéhez add hozzá a következő sorokat:

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

Plug 'kaarmu/typst.vim'

call plug#end()

A szakasz a `call plug#begin()` paranccsal indul, a `Plug` paranccsal kilistázza a kiterjesztéseket és a `call plug#end()` paranccsal zárul.

A 'kaarmu/typst.vim' beírás meghatározza az aktuális [typs.vim](#) kiterjesztést. A név (fejlesztő) + tároló nevesémát követi.

```
david@pclos-mate:~  
File Edit View Search Terminal Help  
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  
15  
16 " Plugin section  
17 call plug#begin()  
18  
19 Plug 'kaarmu/typst.vim'  
20  
21 call plug#end()
```

Mentsd a változtatásaidat és lépj ki a Vim-ből.

Később más kiterjesztéseket is hozzáadhatsz, ha kell. A további bejegyzéseket "**plug#begin**" és "**plug#end**" sorok közé kell elhelyezni. A kiterjesztések bejegyzései a korábban leírt névadási konvenciót kell kövessék!

A typst.vim kiterjesztés telepítése

Indítsd újra a Vim-et: **\$ vim**

Parancssor-módban írd be: **:Plugininstall**

(Megnyitja Vim-ben a kiterjesztéskezelőt és telepíti a .vimrc fájlban felsorolt összes kiegészítőt. A telepített kiterjesztések a következő Vim induláskor automatikusan betöltődnek.)

```
~      Help poor children in Uganda!
~ type :help iccf<Enter>           for inf
~
~ type :q<Enter>                   to exit
~ type :help<Enter> or <F1>       for on-
~ type :help version9<Enter>     for ver
~
~
~
~
~
```

[No Name]

:PlugInstall

Siker esetén a következő képernyőképhez hasonlót kell látnod:

```
david@p
File Edit View Search Terminal Help
[Plugins] [No Name]
1 Updated. Elapsed time: 0.593683 sec.
2 [=]
3
4 - Finishing ... Done!
5 - typst.vim: Already up to date.

[Plugins]
```

További plugin parancsok

Ez a három parancs hasznodra válhat; szintén a Vim parancsmódjában kell kiadni ezeket:

:PlugUpdate — kiterjesztés telepítése, vagy frissítése;

:PlugDiff — az előző frissítéshez képesti változások megtekintése;

:PlugClean — a listában már nem szereplő kiterjesztések eltávolítása.

A typst.vim kiterjesztés alaphasználata

:TypstWatch

A helyi Typst-fájl megnyitása és a szerkesztés elkezdése.

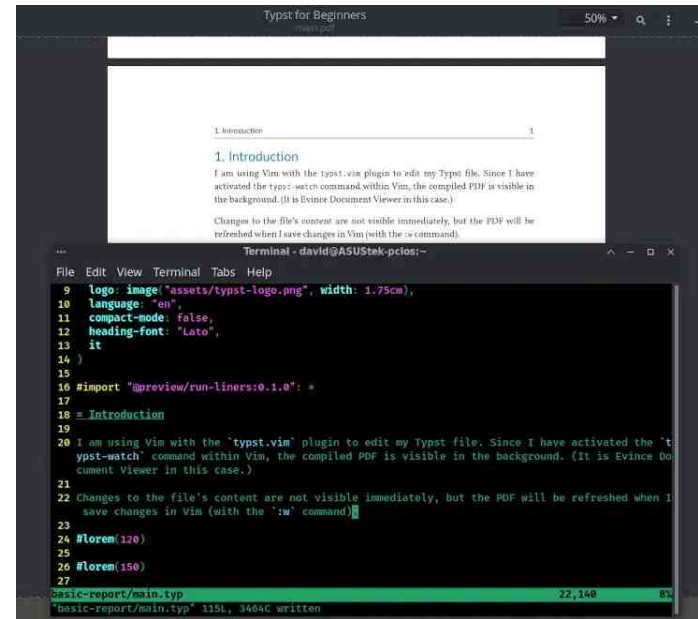
Parancs-módban írd be: **:TypstWatch**

A **typst-watch** parancs aktiválja a Typst olyan módját, amiben folyamatosan figyeli egy adott fájl változásait. Amikor mented a változtatásokat, a Typst újra lefordítja a fájlt és frissíti a megjelenített PDF-et.

A parancs megnyitja a lefordított PDF-et egy olvasóban – Evince dokumentummegjelenítő, a lenti képernyőképen.

A PDF-olvasót a háttérben nyitva hagyhatod. A forrásfájl változásainak mentése („:w” parancssal) után a Typst automatikusan frissíti a PDF-et.

Ha több PDF-olvasód is van és lecserélnéd az alapbeállítás szerintit, köz-zé tettem egy [projektet](#) a feladat KDE Plasma, XFCE és MATE környezetben történő megvalósítására való tippekkel [6.1.1. szakasz, 6-8. o.].

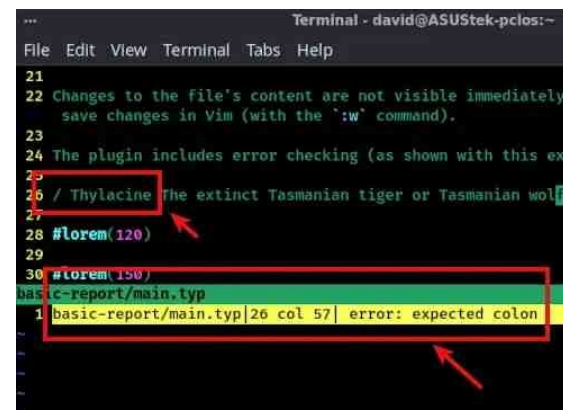


Hibaellenőrzés

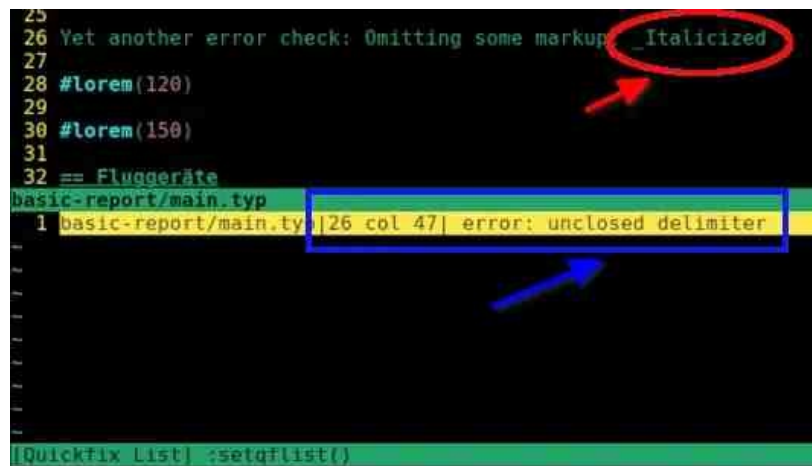
A **typst.vim** kiterjesztés hibaellenőrzéssel is rendelkezik.

A következő példában szándékosan elhagytam a „*Thylacine*” végéről a szükséges kettőspontot a [terminológialistában](#) („ / *Thylacine*: ” kellene).

A fájlom mentésekor a kiterjesztés jelezte, hogy hiba van:



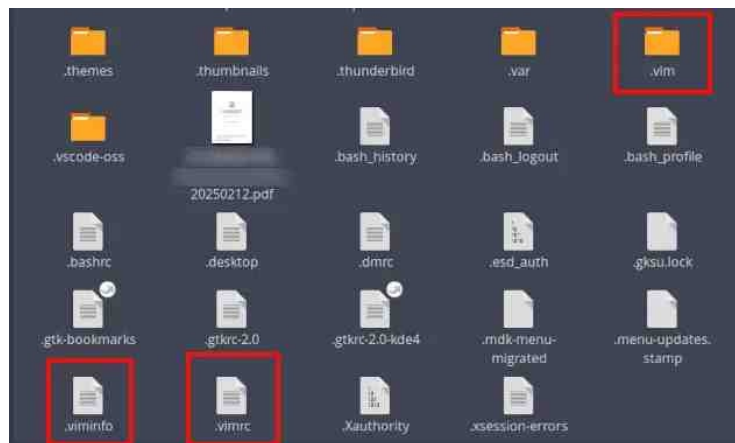
A második példában lent, szándékosan elhagytam a szükséges záró aláhúzást ami az „*Italicized*”-hoz kell („*Italicized*” kellene). A kiterjesztés riaszt a hiba felől.



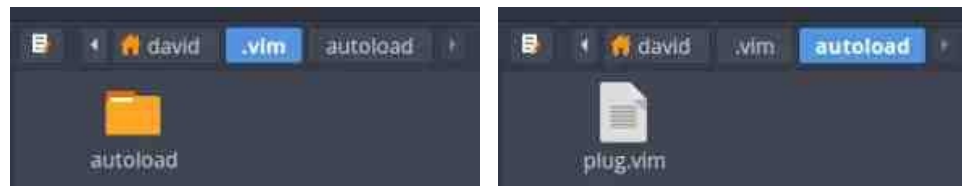
A Vim könyvtárszerkezete

Ha érdekel, íme egy kép arról, hogyan néz ki a Vim-hez kapcsolódó könyvtárszerkezet a **vim-plug** kiterjesztéskezelő és a **typst.vim** kiterjesztés telepítése után:

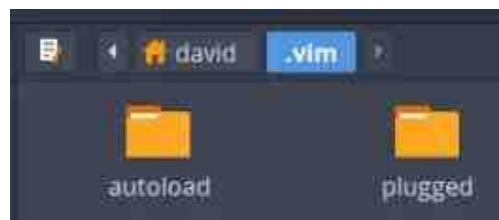
A /home könyvtárban lesz egy **.vimrc** és egy **.viminfo** fájl.



Lesz még egy **.vim** könyvtár is, különféle alkönyvtárakkal. Az **autoload** alkönyvtár tartalmazza a **plug.vim** fájlt.



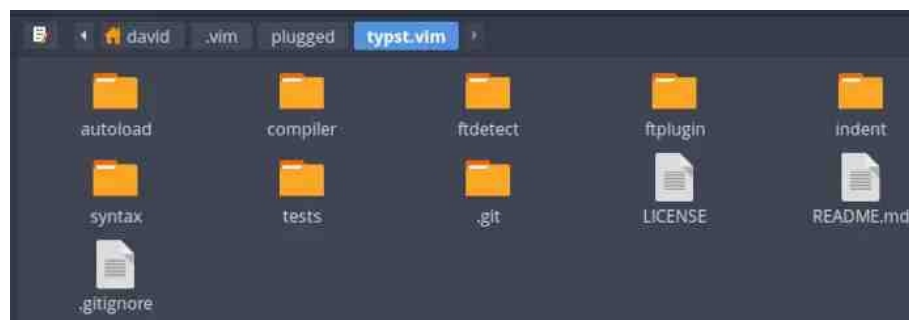
Kell lennie még egy **plugged** nevű **.vim** alkönyvtárnak is:



Ebben az alkönyvtárban van a **typst.vim** alkönyvtár:



Ugyanakkor, a **typst.vim** alkönyvtár tartalmazza a kiterjesztés telepítése közben kaarmu GitHub-tárolójából letöltött különféle részeket.



Legyen élvezetes a Typst-ben a Vim használata szerkesztőként!