

TeXnical Vim

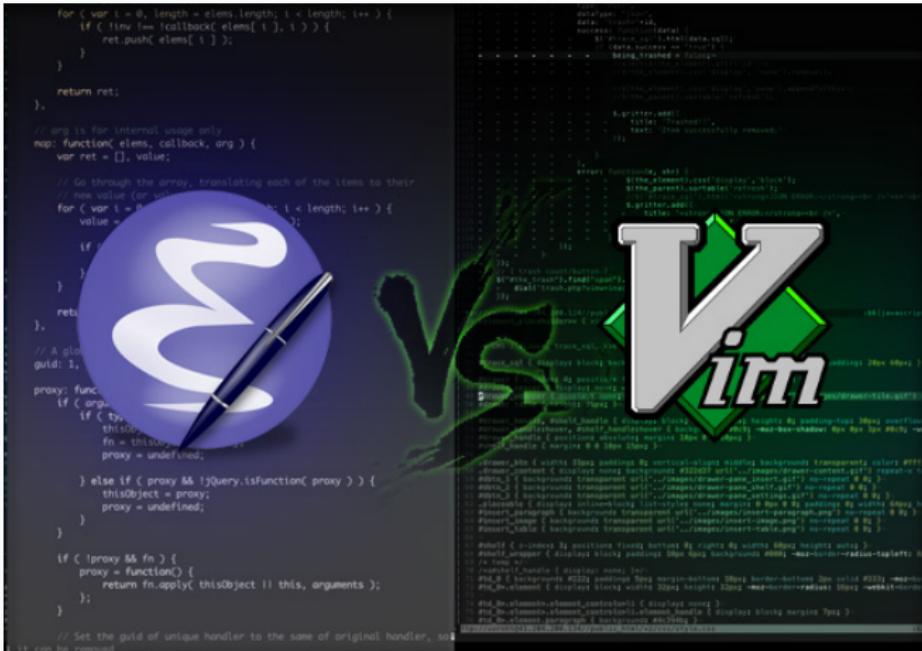
이재호

2020년 2월 15일

서울대학교 전기·정보공학부 / KTUG



Editor War (1985~현재)



그렇다면 어떤 에디터를 쓰는게 제일 좋을까요?

그렇다면 어떤 에디터를 쓰는게 제일 좋을까요?

여러분의 취향에 맞는 에디터를 쓰세요!

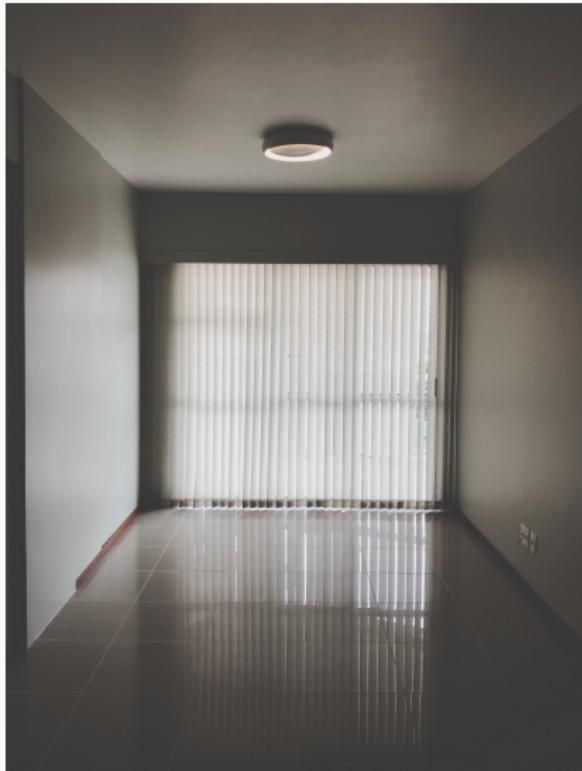
그렇다면 어떤 에디터를 쓰는게 제일 좋을까요?

여러분의 취향에 맞는 에디터를 쓰세요!

한 가지 취향과 가능성을 소개해드리려고 합니다.

Vim의 가장 큰 무기는 원하는대로 만들 수 있다는 것입니다.

Vim의 가장 큰 무기는 원하는대로 만들 수 있다는 것입니다.



Vim의 가장 큰 무기는 원하는대로 만들 수 있다는 것입니다.

그러나, 이는 동시에
진입장벽이기도 합니다.



Vim의 가장 큰 무기는 원하는대로 만들 수 있다는 것입니다.



그러나, 이는 동시에
진입장벽이기도 합니다.

```
!TEX program = xelatex
\documentclass{beamer}

\usepackage{lipsum}
\usepackage{tikz}
\usepackage{graphicx}
\usepackage{adjustbox}
\usepackage{textcomp}

\vfuzz=3pt

%%%%%%%%%%%%%
% Beamer Settings %
%%%%%%%%%%%%%
\usepackage[numbering=fraction]{metropolis}
\usecolortheme{rose}
\useoutertheme{subsection=false}[miniframes]

\setbeamertemplate{itemize item}{\square}
\setbeamertemplate{itemize subitem}{\triangle}
\setbeamertemplate{itemize subsubitem}{\circle*}

%%%%%%%%%%%%%
% Font Settings %
%%%%%%%%%%%%%
\usepackage[math-style=ISO]{mathtools}
\usepackage[utf8]{inputenc}
\usepackage{fontspec}
\usepackage{mathrsfs} [unicode-math]
\usepackage{mathrsfs} [unicode-math]

\setsansfont{Roholot}
\boldfont = +Medium;
\bolditalicfont = +MediumItalic
\setsansfont{Inconsolata}
\setsansfont{NuumGothic} [AutoFakeSlant=0.18]

%%%%%%%%%%%%%
% Minted Settings %
%%%%%%%%%%%%%
\renewcommand{\fancyVerbLine}{\textsf{\tiny arabic}\fancyVerbLine}})

\newenvironment{lateX}[1]{
    \escapeinside{|`|},
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
        \node [inner sep=0pt, right=10pt] (output) {\texttt{`#1`}};
        \draw [line width=1pt, ->] (code) --> (output);
    \end{tikzpicture}
}{\end{tikzpicture} \end{lateX}

\newenvironment{linetikz}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{linetikz}

\newenvironment{listtikz}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{listtikz}

\newenvironment{scriptstikz}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{scriptstikz}

\newenvironment{scriptsize}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{scriptsize}

\newenvironment{tiny}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tiny}

\newenvironment{arabic}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{arabic}

\newenvironment{fancyVerbLine}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{fancyVerbLine}

\newenvironment{tinyarabic}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinyarabic}

\newenvironment{tinyfancyVerbLine}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinyfancyVerbLine}

\newenvironment{scriptsizearabic}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{scriptsizearabic}

\newenvironment{scriptsizefancyVerbLine}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{scriptsizefancyVerbLine}

\newenvironment{tinyfancyVerbLine}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinyfancyVerbLine}

\newenvironment{tinyarabic}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinyarabic}

\newenvironment{tinyfancyVerbLine}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinyfancyVerbLine}

\newenvironment{scriptsizefancyVerbLine}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{scriptsizefancyVerbLine}

\newenvironment{tinyscriptsize}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinyscriptsize}

\newenvironment{tinytiny}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinytiny}

\newenvironment{tinytinyarabic}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinytinyarabic}

\newenvironment{tinytinyfancyVerbLine}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinytinyfancyVerbLine}

\newenvironment{tinytinyfancyVerbLine}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinytinyfancyVerbLine}

\newenvironment{tinytinyarabic}[1]{
    \begin{tikzpicture}[remember picture, overlay]
        \node [inner sep=0pt] (code) {\texttt{#1}};
    \end{tikzpicture}
}{\end{tikzpicture} \end{tinytinyarabic}
```

가능성

Technical-Vim

I II III IV

texnical-vim.tex

```
1: texnical-vim.tex >
  h Toggle help text
  t Toggle sorted TODOs
  +/- Decrease/increase ToC depth
  f/F Apply/clear filter
  s Hide numbering

Layers: L label
        I include
        T todo
        C content

Preamble
Frame
Frame
Frame
Frame

3 Why Vim?
Frame: 첫 번째 관문
Frame: 본 발표는...
Frame: 그러나 본 발표는...
Frame: Vim이 뭘까요?
Frame: 제 X 대신 Vim인가요?
Frame: (Neo)Vim의 뜻 모습
Frame: \TeXnical Vim 설정
Frame

Table of contents (vintex) -> texnical-vim.tex      tex > (c) 428W < 51% ≡ 119/230 ≈ : 36
1 texnical-vim.tex[173 warning] Overfull \vbox (377.6639pt too high) detected at line 173$
```

Quickfix Vintex errors (LaTeX logfile)

```
:resize +5
[tex] @:nvim* 1:nvim-
```

buffers

imux

66% 21:48

texnical-vim.pdf (page 2 of 16)

Previous Next Page Back/Forward Zoom Tool Mode

TExnical Vim

이제호
2020년 2월 15일
서울대학교 전기·정보공학부/KTUG

MyPhygmyed
MyPhygmyed

manus

TeXnical Vim

Atom

B

Emacs

Sublime Text

Notepad++

VS Code

gEdit

TeXShop

TeX

Vim

VS Code

Page 2 of 16

본 발표는...

T_EX 사용자 중

1. Vim에 대해서 들어봤지만 시도하기가 막막하셨던 분들,

본 발표는...

T_EX 사용자 중

1. Vim에 대해서 들어봤지만 시도하기가 막막하셨던 분들,
2. Vim을 쓸 줄 알지만 자료가 별로 없어서 T_EX을 Vim에서 쓰지 않았던 분들,

본 발표는...

T_EX 사용자 중

1. Vim에 대해서 들어봤지만 시도하기가 막막하셨던 분들,
2. Vim을 쓸 줄 알지만 자료가 별로 없어서 T_EX을 Vim에서 쓰지 않았던 분들,
3. Vim(혹은 Emacs+AUCT_EX)으로 T_EX을 쓰지만 다른 사람들의 작업 환경이 궁금하신 분들

본 발표는...

T_EX 사용자 중

1. Vim에 대해서 들어봤지만 시도하기가 막막하셨던 분들,
2. Vim을 쓸 줄 알지만 자료가 별로 없어서 T_EX을 Vim에서 쓰지 않았던 분들,
3. Vim(혹은 Emacs+AUCT_EX)으로 T_EX을 쓰지만 다른 사람들의 작업 환경이 궁금하신 분들

Vim으로 T_EX 문서를 작성하고 편집하는 효율적인 세팅!

Why Vim?

Vim Setup

Hello, Vim!

TEXnical Vim

Why Vim?

첫 번째 관문

T_EX을 처음 깔았을 때 누구나 마주하는 문제:

첫 번째 관문

*T_EX*을 처음 깔았을 때 누구나 마주하는 문제:

T_EX 문서를 어떤 도구로 만들까?

첫 번째 관문

*T_EX*을 처음 깔았을 때 누구나 마주하는 문제:

T_EX 문서를 어떤 도구로 만들까?

가장 기본적인 선택지:

- *T_EXworks* (Windows)
- *T_EXShop* (macOS)

Why Vim?
○○●○○○

Vim Setup
○○○

Hello, Vim!
○

TExnical Vim
○○○

Vim이 뭔가요?

Vim이 뭔가요?

Vim은 modal(모달) 에디터입니다.

Vim이 뭔가요?

Vim은 modal(모달) 에디터입니다.

- Normal, Insert, Visual, Command-line 모드
 - ▶ 와 Ex, Select 모드

Vim이 뭔가요?

Vim은 modal(모달) 에디터입니다.

- Normal, Insert, Visual, Command-line 모드
 - ▶ 와 Ex, Select 모드
- Editor라는 이름에 걸맞게, 편집하는 일에 특화
 - ▶ 반복 작업 및 영역 편집

Vim이 뭔가요?

Vim은 modal(모달) 에디터입니다.

- Normal, Insert, Visual, Command-line 모드
 - ▶ 와 Ex, Select 모드
- Editor라는 이름에 걸맞게, 편집하는 일에 특화
 - ▶ 반복 작업 및 영역 편집
- 기본적으로는 터미널에서 사용
 - ▶ 손은 키보드 위에서만

Vim이 뭔가요?

Vim은 modal(모달) 에디터입니다.

- Normal, Insert, Visual, Command-line 모드
 - ▶ 와 Ex, Select 모드
- Editor라는 이름에 걸맞게, 편집하는 일에 특화
 - ▶ 반복 작업 및 영역 편집
- 기본적으로는 터미널에서 사용
 - ▶ 손은 키보드 위에서만
- 매우 유연한 설정
 - ▶ 그만큼 초기 설정은 매우 적음

왜 X 대신 Vim인가요?

- X = Electron 계열의 에디터 (Atom, Visual Studio Code)

왜 X 대신 Vim인가요?

- X = Electron 계열의 에디터 (Atom, Visual Studio Code)
 - ▶ 장점
 - GUI로 동작하므로 진입장벽이 낮음

왜 X 대신 Vim인가요?

■ X = Electron 계열의 에디터 (Atom, Visual Studio Code)

▶ 장점

- GUI로 동작하므로 진입장벽이 낮음
- Language server를 통한 강력한 문법 강조 및 자동 완성

왜 X 대신 Vim인가요?

■ X = Electron 계열의 에디터 (Atom, Visual Studio Code)

▶ 장점

- GUI로 동작하므로 진입장벽이 낮음
- Language server를 통한 강력한 문법 강조 및 자동 완성

▶ 단점

- (텍스트 편집을 위해서!) 웹 브라우저(Chromium)의 부하

왜 X 대신 Vim인가요?

■ X = Electron 계열의 에디터 (Atom, Visual Studio Code)

▶ 장점

- GUI로 동작하므로 진입장벽이 낮음
- Language server를 통한 강력한 문법 강조 및 자동 완성

▶ 단점

- (텍스트 편집을 위해서!) 웹 브라우저(Chromium)의 부하
- Language Server Protocol (LSP)의 등장

왜 X 대신 Vim인가요?

■ X = Electron 계열의 에디터 (Atom, Visual Studio Code)

▶ 장점

- GUI로 동작하므로 진입장벽이 낮음
- Language server를 통한 강력한 문법 강조 및 자동 완성

▶ 단점

- (텍스트 편집을 위해서!) 웹 브라우저(Chromium)의 부하
- Language Server Protocol (LSP)의 등장

■ X = Emacs

왜 X 대신 Vim인가요?

■ X = Electron 계열의 에디터 (Atom, Visual Studio Code)

▶ 장점

- GUI로 동작하므로 진입장벽이 낮음
- Language server를 통한 강력한 문법 강조 및 자동 완성

▶ 단점

- (텍스트 편집을 위해서!) 웹 브라우저(Chromium)의 부하
- Language Server Protocol (LSP)의 등장

■ X = Emacs

▶ 장점

- CLI로 동작함

왜 X 대신 Vim인가요?

■ X = Electron 계열의 에디터 (Atom, Visual Studio Code)

▶ 장점

- GUI로 동작하므로 진입장벽이 낮음
- Language server를 통한 강력한 문법 강조 및 자동 완성

▶ 단점

- (텍스트 편집을 위해서!) 웹 브라우저(Chromium)의 부하
- Language Server Protocol (LSP)의 등장

■ X = Emacs

▶ 장점

- CLI로 동작함
- More than an editor

왜 X 대신 Vim인가요?

■ X = Electron 계열의 에디터 (Atom, Visual Studio Code)

▶ 장점

- GUI로 동작하므로 진입장벽이 낮음
- Language server를 통한 강력한 문법 강조 및 자동 완성

▶ 단점

- (텍스트 편집을 위해서!) 웹 브라우저(Chromium)의 부하
- Language Server Protocol (LSP)의 등장

■ X = Emacs

▶ 장점

- CLI로 동작함
- More than an editor

▶ 단점

왜 X 대신 Vim인가요?

■ X = Electron 계열의 에디터 (Atom, Visual Studio Code)

▶ 장점

- GUI로 동작하므로 진입장벽이 낮음
- Language server를 통한 강력한 문법 강조 및 자동 완성

▶ 단점

- (텍스트 편집을 위해서!) 웹 브라우저(Chromium)의 부하
- Language Server Protocol (LSP)의 등장

■ X = Emacs

▶ 장점

- CLI로 동작함
- More than an editor

▶ 단점

- CLI로 동작함
- More than an editor

2020년의 Vim

- User Friendly!

2020년의 Vim

■ User Friendly!

- ▶ LSP의 혜택으로 강력한 문법 강조, 자동 완성, 문법 검증

2020년의 Vim

■ User Friendly!

- ▶ LSP의 혜택으로 강력한 문법 강조, 자동 완성, 문법 검증
 - LSP가 너무 무겁다면, 기존 방식의 문법팩 사용 가능
- ▶ 모든 것이 사용자화 가능

2020년의 Vim

■ User Friendly!

- ▶ LSP의 혜택으로 강력한 문법 강조, 자동 완성, 문법 검증
 - LSP가 너무 무겁다면, 기존 방식의 문법팩 사용 가능
- ▶ 모든 것이 사용자화 가능
- ▶ 직관적이고 편한 명령어 (동사 + 명사)

2020년의 Vim

■ User Friendly!

- ▶ LSP의 혜택으로 강력한 문법 강조, 자동 완성, 문법 검증
 - LSP가 너무 무겁다면, 기존 방식의 문법팩 사용 가능
- ▶ 모든 것이 사용자화 가능
- ▶ 직관적이고 편한 명령어 (동사 + 명사)
- ▶ 40년이 넘는 자료와 사용자 그룹

2020년의 Vim

■ User Friendly!

- ▶ LSP의 혜택으로 강력한 문법 강조, 자동 완성, 문법 검증
 - LSP가 너무 무겁다면, 기존 방식의 문법팩 사용 가능
- ▶ 모든 것이 사용자화 가능
- ▶ 직관적이고 편한 명령어 (동사 + 명사)
- ▶ 40년이 넘는 자료와 사용자 그룹
- ▶ 한 가지 일을 하고, 그것을 잘함 (UNIX Philosophy: Do One Thing and Do It Well)

2020년의 Vim

■ User Friendly!

- ▶ LSP의 혜택으로 강력한 문법 강조, 자동 완성, 문법 검증
 - LSP가 너무 무겁다면, 기존 방식의 문법팩 사용 가능
- ▶ 모든 것이 사용자화 가능
- ▶ 직관적이고 편한 명령어 (동사 + 명사)
- ▶ 40년이 넘는 자료와 사용자 그룹
- ▶ 한 가지 일을 하고, 그것을 잘함 (UNIX Philosophy: Do One Thing and Do It Well)

■ Developer Friendly!

2020년의 Vim

■ User Friendly!

- ▶ LSP의 혜택으로 강력한 문법 강조, 자동 완성, 문법 검증
 - LSP가 너무 무겁다면, 기존 방식의 문법팩 사용 가능
- ▶ 모든 것이 사용자화 가능
- ▶ 직관적이고 편한 명령어 (동사 + 명사)
- ▶ 40년이 넘는 자료와 사용자 그룹
- ▶ 한 가지 일을 하고, 그것을 잘함 (UNIX Philosophy: Do One Thing and Do It Well)

■ Developer Friendly!

- ▶ Lua를 내장한 Neovim의 등장으로 Lua로 스크립팅이 가능해짐 (LuaTeX?)

2020년의 Vim

■ User Friendly!

- ▶ LSP의 혜택으로 강력한 문법 강조, 자동 완성, 문법 검증
 - LSP가 너무 무겁다면, 기존 방식의 문법팩 사용 가능
- ▶ 모든 것이 사용자화 가능
- ▶ 직관적이고 편한 명령어 (동사 + 명사)
- ▶ 40년이 넘는 자료와 사용자 그룹
- ▶ 한 가지 일을 하고, 그것을 잘함 (UNIX Philosophy: Do One Thing and Do It Well)

■ Developer Friendly!

- ▶ Lua를 내장한 Neovim의 등장으로 Lua로 스크립팅이 가능해짐 (LuaTeX?)
- ▶ 커뮤니티 기반 (무엇이 Vim이고 Neovim일까요?)



Why Vim?
○○○○●

Vim Setup
○○○

Hello, Vim!
○

TExnical Vim
○○○



neovim

Vim Setup

TeXnical Vim 설정 (for macOS)

macOS용 package manager인 Homebrew 사용

1. <https://brew.sh/>를 참고하여 Homebrew를 설치합니다.
2. iTerm2, MacTeX, Skim을 설치합니다.

```
brew tap homebrew/cask
brew cask install item2 mactex skim
```

iTerm2 Color scheme, option key mapping,
powerline glyphs, ...

MacTeX TeXLive의 macOS용 재배포

Skim SyncTeX support, automatic refresh, ...

TExnical Vim 설정 (for Windows)

Windows용 package manager인 Chocolatey 사용

1. <https://chocolatey.org/install>을 참고하여 Chocolatey를 설치합니다. (명령어가 너무 길어 링크로 대체)
2. KTUG Wiki를 참고하여, install-tl.zip의 압축을 풀고 install-tl-windows.bat을 실행하여 설치합니다.
3. `choco install microsoft-windows-terminal neovim
→ sumatrapdf`

Windows Terminal Color scheme (that supports iTerm themes!), **option key mapping**, powerline glyphs, ...

SumatraPDF SyncTeX support, automatic refresh, ...

Hello, Vim!

TeXnical Vim

Why Vim?
○○○○○

Vim Setup
○○○

Hello, Vim!
○

TExnical Vim
○●○

감사합니다.

부록

발표 자료:

<https://github.com/Zeta611/txnical-vim-kts-conf-2020>

Neovim 설정: <https://github.com/Zeta611/dotfiles>

