

MiKTeX으로 구축하는 작업 환경

Nova de Hi likesam aud

2023년 2월 28일

version 1.4.1

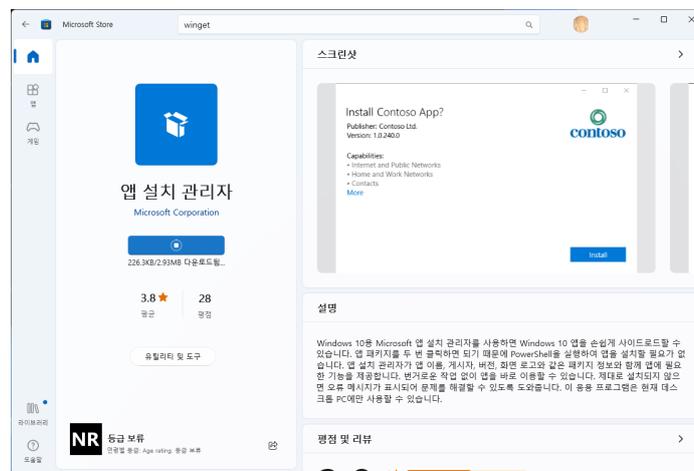
준비

1 winget 설치

이미 winget을 사용할 수 있다면 이 단계를 건너뛸다.

winget은 Windows 패키지 설치 관리자이다. 이후의 설명은 winget을 이용하여 설치하는 예를 보일 것이므로 winget이 사용가능하도록 해두어야 한다. winget은 Windows 10, Windows 11에서 설치 가능하다.

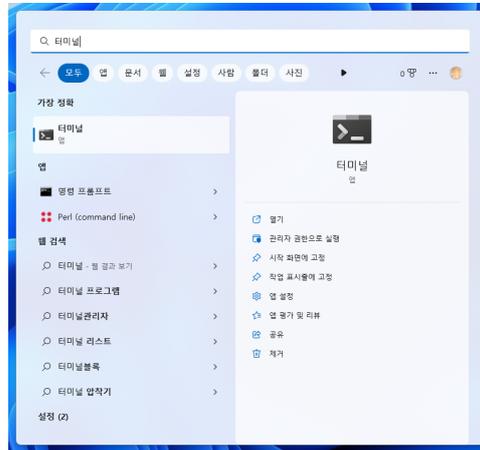
- Microsoft Store를 열어서 winget으로 검색한다.
- 그 가운데 ‘**앱 설치 관리자**’를 선택하여 활성화한다.



2 터미널 열기

2.1 터미널 앱

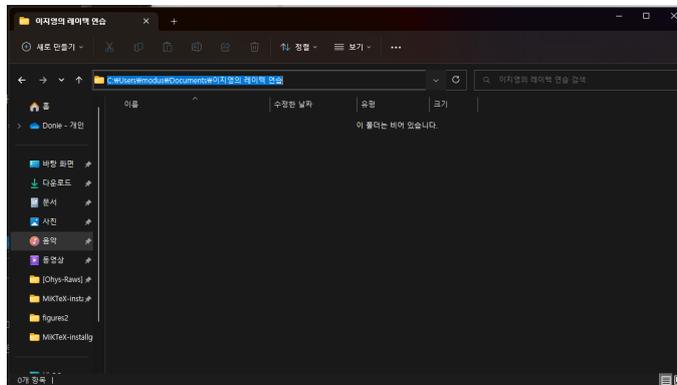
Windows에서 터미널을 열 수 있는지 확인한다. 터미널이라는 앱은 Windows 11에는 기본으로 설치되어 있고, Windows 10이라면 Microsoft Store에서 검색하여 설치할 수 있다.

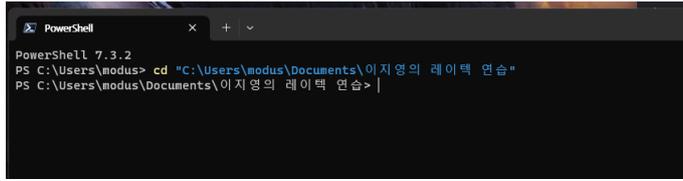


참고 터미널이 Windows 11의 기본 앱이기 때문에 이 안내서는 터미널을 중심으로 설명한다. Windows에는 예전부터 cmd.exe라는 명령행 인터페이스가 있었기 때문에 터미널을 별도로 설치해야 한다면 그냥 cmd를 쓰는 것으로 충분하므로, 이후 터미널을 열라는 설명은 cmd라고 간주해도 무방하다.

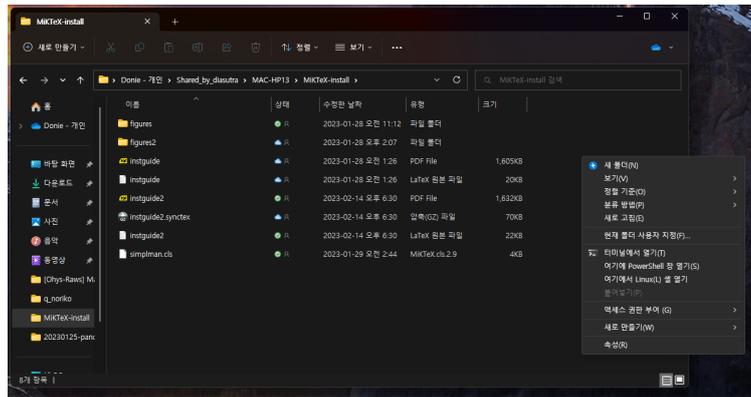
2.2 현재 폴더로 터미널 열기

Windows 탐색기로 어떤 폴더를 열어둔 상태에서, 현재 폴더 위치로 터미널을 열어야 할 때가 있다. **Ctrl** 키를 눌러서 터미널을 실행하면 %userprofile%이라고 하는, 사용자의 홈 폴더 위치로 터미널이 열린다. 이 위치로부터 cd 명령을 사용하여 원하는 위치까지 이동할 때, 탐색기의 주소표시줄을 눌러서 폴더의 Path를 얻어내고(**Ctrl**+**C**) 이것을 cd 명령의 인자로 붙여넣기(**Ctrl**+**V**)하면 간단하다. 이 때 주의하여야 하는 것은 복사된 Path에 빈 칸이 포함되어 있으면 따옴표로 둘러싸주어야 하는 것이다.





또다른 방법으로 현재 열린 탐색기 폴더 창의 빈 공간에서 [Shift]+ [Right Click]하여 “터미널에서 열기”를 찾아보는 방법이 있다.



두 방법 모두 익숙해지는 것이 좋다.

MiKTeX 설치

1 winget으로 설치

터미널을 열고 다음 명령을 내려 MiKTeX이 설치 가능한지 확인한다.

```
winget search MiKTeX
```

winget을 처음 사용한다면 다음과 같은 화면이 나오는데 를 입력하여 진행한다.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치하세요! https://aka.ms/PSWindows

PS C:\Users\diasu> winget search MiKTeX
'msstore' 원본을 사용하려면 다음 계약을 확인해야 합니다.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
원본이 제대로 작동하려면 현재 컴퓨터의 두 글자 지리적 지역을 백 엔드 서비스로 보내야 합니다(예: "미국").

모든 원본 사용 약관에 동의하십니까?
[Y] 예 [N] 아니요: |
```

MiKTeX을 검색한 결과가 나타난다. 여기서 확인한 ID로 install 명령을 내리면 된다.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

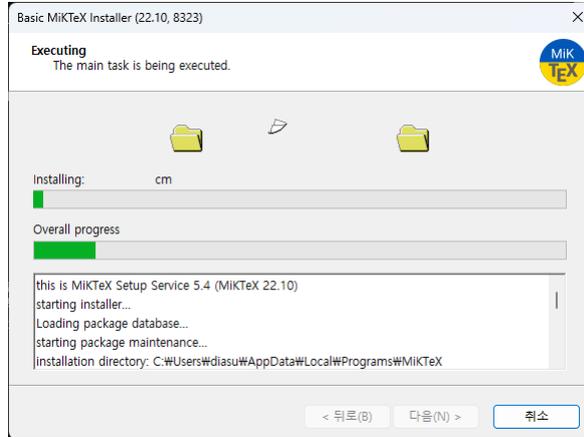
새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치하세요! https://aka.ms/PSWindows

PS C:\Users\diasu> winget search MiKTeX
'msstore' 원본을 사용하려면 다음 계약을 확인해야 합니다.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
원본이 제대로 작동하려면 현재 컴퓨터의 두 글자 지리적 지역을 백 엔드 서비스로 보내야 합니다(예: "미국").

모든 원본 사용 약관에 동의하십니까?
[Y] 예 [N] 아니요: y
이름   장치 ID   버전   일지   원본
-----
MiKTeX MiKTeX.MiKTeX 22.10 ProductCode: miktex_winget
PS C:\Users\diasu> winget install MiKTeX.MiKTeX
```

```
winget install MiKTeX.MiKTeX
```

다음과 같은 화면이 나타나면서 설치가 진행된다. 설치가 진행 중일 때에 아무 것도 건드릴 필요가 없다.



완료되면 **종료** 버튼을 누른다. 이것으로 MiKTeX 설치는 완료되었다.

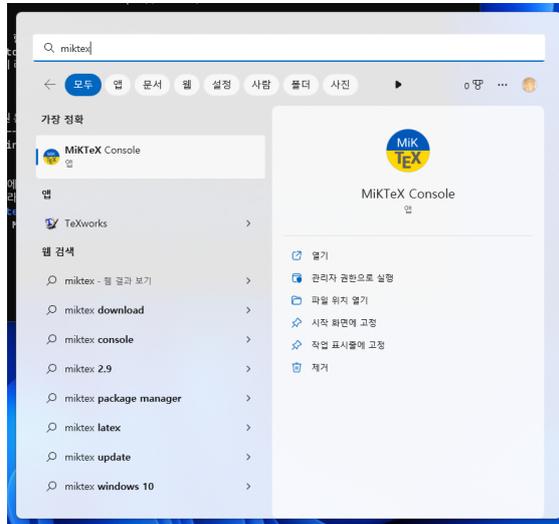
참고 `winget`을 이용하지 않고 MiKTeX 설치 프로그램을 이용할 수 있다. miktex.org/Download 페이지에서 설치 프로그램을 다운로드하여 실행하면 비슷한 과정을 거쳐 설치할 수 있다. 이 때에는 '전체 사용자(for all users)'를 위한 설치가 가능한데 일반적으로 사용자 자신(only for me)을 위하여 설치하는 것이 관리하기 좋다.

2 설치 후

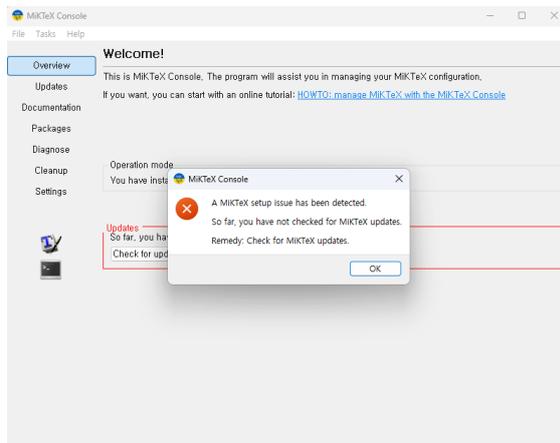
일반적으로 업데이트는 한 달에 한 번 정도 하는 것을 추천한다. 최초에는 필수적으로 업데이트하여야 한다.

MiKTeX의 설치 후에는 update를 한 번 해주어야 한다.

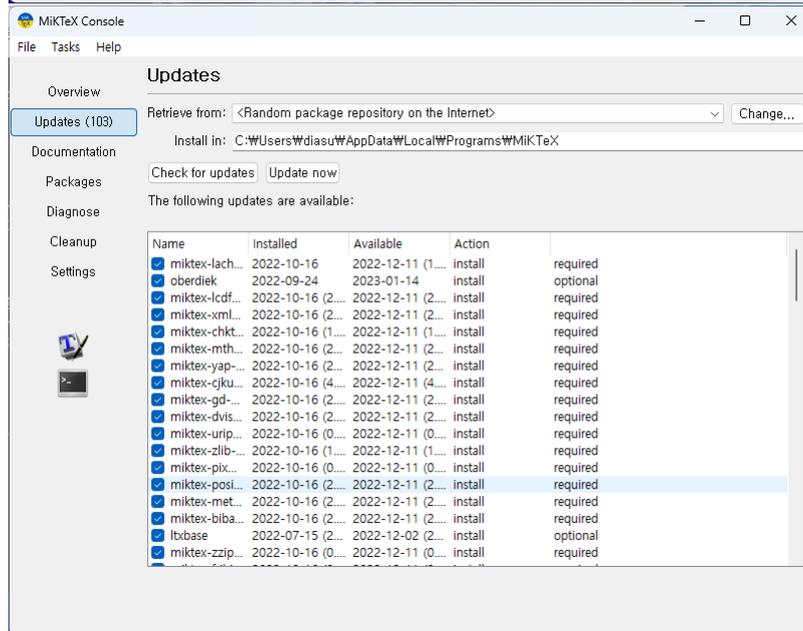
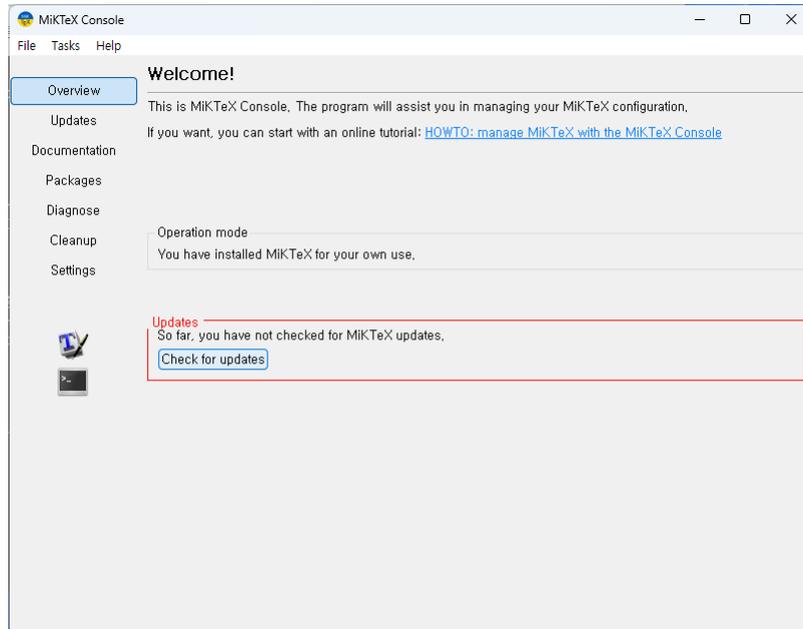
Windows 키 (⊞)를 눌러서 MiKTeX-Console이라는 프로그램을 찾아 실행한다.



실행된 상태에서 맨처음에 다음과 같은 경고 메시지가 나오는데, 그냥 OK를 눌러서 진행하면 된다.



여기서 “Check for updates”라는 버튼을 누르고 업데이트 체크가 끝나기를 기다린다. 왼쪽 메뉴의 Updates 항목으로 이동하여 **Update now** 버튼을 누른다.



업데이트가 종료되면 “MiKTeX Console needs to be closed”라는 메시지가 나타나는데 OK를 누르면 MiKTeX-Console의 실행이 종료된다. 업데이트가 잘 완료된 것이다.

참고 업데이트가 성공적으로 진행되지 않거나 뭔가 문제가 있다는 메시지가 나타나는 것은 MiKTeX 서버에서 업데이트가 진행중이거나 미리 서버가 동작하지 않거나 인터넷에 연결되어 있지 않기 때문이다. 이럴 때는 몇 시간 기다렸다가 재시도한다.

3 Perl 설치(거의 필수)

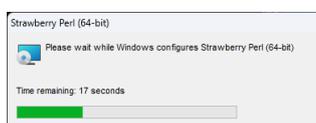
이 상태로도 MiKTeX을 운영하는 데 큰 문제는 없으나 Perl이 설치되어 있으면 latexmk나 pdfcrop, komindex와 같은 필수 유틸리티를 실행할 수 있다.

Perl은 다음 명령으로 설치한다.

```
winget install StrawberryPerl.StrawberryPerl
```

```
PS C:\Users\diasu> winget search StrawberryPerl
이름      장치 ID      버전      일치      원본
-----
Strawberry Perl StrawberryPerl.StrawberryPerl 5.32.1001 Moniker: strawberryperl winget
PS C:\Users\diasu> winget install StrawberryPerl.StrawberryPerl
```

Strawberry Perl 설치 프로그램이 실행되면 ‘사용자 계정 컨트롤’에서 “게시자를 알 수 없는 이 앱이 디바이스를 변경할 수 있도록 허용하시겠습니까?”라고 물어온다. 여기서 “예”를 클릭하고 진행한다.



4 기타 유틸리티(선택)

추가 유틸리티를 설치하면 MiKTeX의 기능을 더욱 풍부하게 사용할 수 있다. 예를 들면 python과 pygments 같은 것이다. 다만 이것이 당장 필수적인 것은 아니며 나중에 필요해 졌을 때 진행하면 되는 것이므로 이에 대하여 부록(31페이지)에 필요한 정보를 적어두었다.

참고 이 안내서는 한국어 문서 작성을 주로 하는 상황을 가정하고 있다. 만약 L^AT_EX을 설치하는 목적이 영어 논문을 쓰기 위해서라면 MiKTeX을 설치한 후 그 상태에서 바로 작업을 시작하면 된다. 그 경우에 이 안내서는 여기가 끝이다. 이후 설명하는 폰트 문제라든가 KTUG 사설저장소 패키지 등은 모두 한국어/한글 문서 작성과 관련있는 것이다.

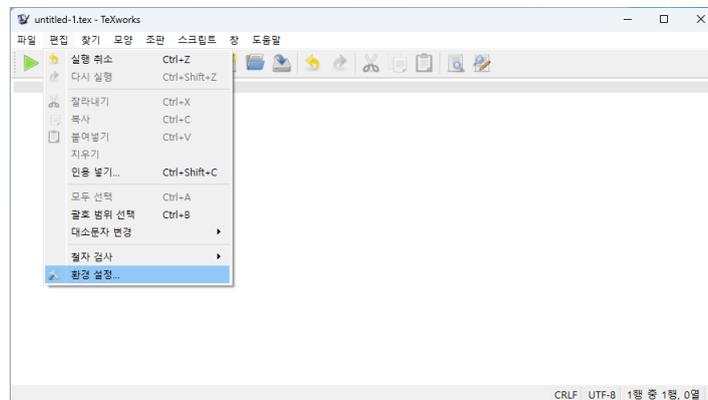
TeXworks 설정

TeX 문서를 편집하기 위한 에디터는 여러 종류가 있으므로 자신의 취향에 맞게 하여 쓰는 것이 가장 좋다. 다른 선택의 여지가 없다면 MiKTeX이 기본 제공하는 TeXworks 에디터를 쓸 수 있다.

1 Preferences

먼저 몇 가지 설정을 해두어야 한다. 우리는 주로 한글 문서를 편집할 것이고 컴파일러 엔진을 Xe_{La}TeX으로 할 것이다. 이를 위하여 다음 몇 가지를 미리 설정해두자. 이 설정은 아무 문서도 편집하지 않는 상태에서 진행하는 것이 좋다.

TeXworks를 실행하여 **편집** > **환경 설정**을 눌러서 환경 설정 창을 연다.

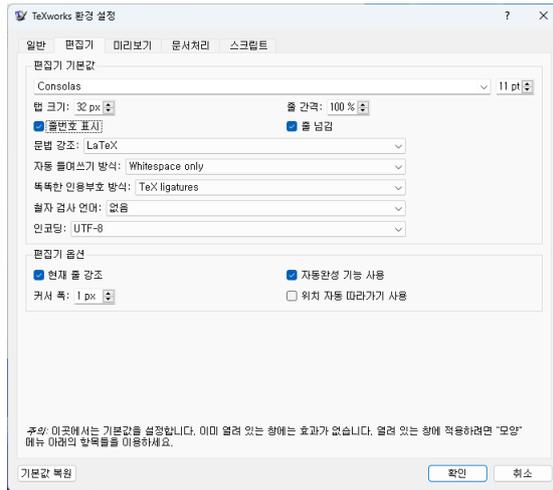


1.1 편집기 탭

편집기 탭에서 다음 몇 가지 항목을 선택한다.

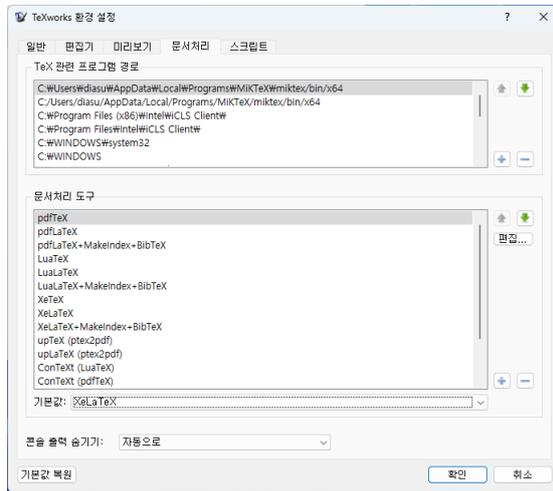
폰트 자신이 선호하는 코딩용 폰트와 크기를 설정한다. 잘 모르겠으면 Consolas 11pt로 하는 것을 권장함.

기타 줄번호 표시, 줄넘김, 현재 줄 강조, 문법 강조=LaTeX



1.2 문서처리 탭

다른 것은 건드리지 말고 ‘기본값’을 XeLaTeX으로 선택한다.



여기서 **확인** 버튼을 누르고 TeXworks 실행을 종료한 다음 다시 실행한다.

첫 문서 처리 및 자동 패키지 설치

MiKTeX의 대표적인 기능 중 하나가 ‘MiKTeX 패키지 관리자(mpm)’를 이용하여 필요한 패키지를 온라인에서 즉시 불러와 설치하는 것이다. TeXLive에서는 이 기능이 없거나 취약하기 때문에 미리 전체 설치를 하도록 권장하고 있는 반면, MiKTeX은 처음에 가벼운 규모의 텍 시스템만을 설치하고 작업 중에 필요한 것은 그때그때 불러오도록 하고 있다.

일단 mpm에 의하여 설치된 패키지는 그 후 설치된 상태가 유지되기 때문에 처음 한 번 설치한 후에는 다시 같은 패키지를 설치하려고 하지 않는다. 그리고 MikTeX-Console에서 업데이트를 지시하면 현재 설치된 패키지에 업데이트가 있을 때 모두 업데이트한다.

따라서 작업 중에는 기계가 항상 온라인에 연결되어 있어야 한다.

1 첫 문서 작성

TeXworks를 실행하여 다음 그림과 같은 문서를 작성하자.

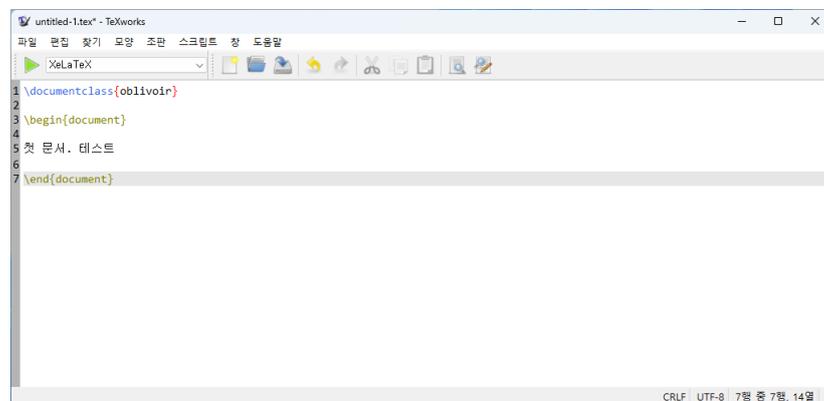
한글이 한 글자 이상 있는 간단한 샘플이면 된다.

```
\documentclass{oblivoir}

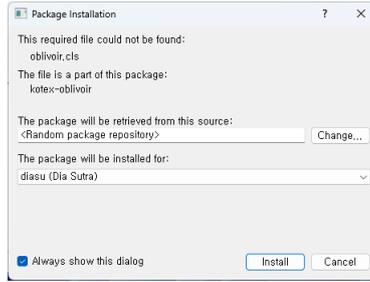
\begin{document}

첫 문서. 테스트

\end{document}
```

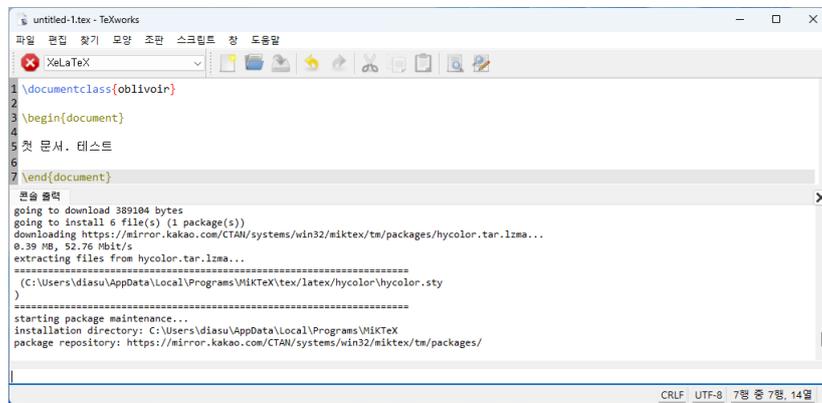


여기서 조판 버튼(▶)을 누르거나 **[Ctrl]+[T]** 단축키를 눌러서 문서의 컴파일을 시도한다. 그러면 (처음 컴파일하는 경우에) 필요한 패키지를 인스톨하기 위하여 다음과 같은 화면이 나타난다.



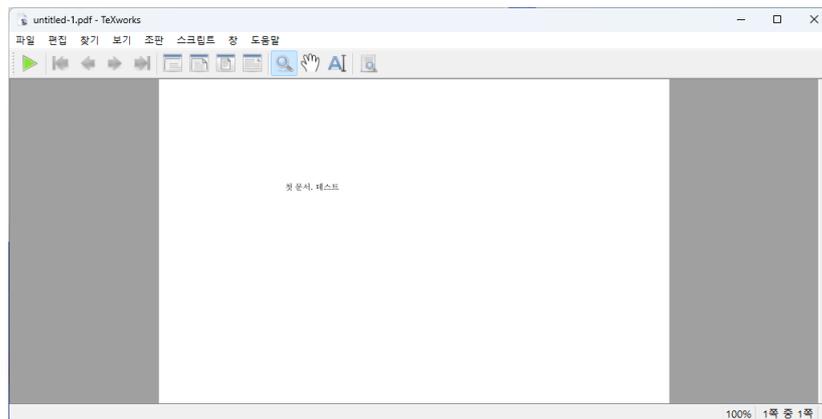
여기서 그냥 **Install**을 누르면 필요한 패키지를 인스톨하고 계속하지만 새로운 패키지에 대하여 같은 화면이 또다시 나타난다. 그러므로 “Always show this dialog” 앞의 체크를 눌러 없애서 MiKTeX이 스스로 필요하면 패키지를 설치하게 하자.

컴파일이 진행되는 동안 편집 화면의 아래쪽에 컴파일 진행 과정이 보인다.



만약 컴파일이 성공하지 못하고 에러스톱이 발생하는 경우에는 컴파일 콘솔 화면의 더 아래에 한 줄짜리 Interactive Input bar가 있으므로 여기에 **x**를 써넣고 **Enter**를 치면 컴파일이 중단된다. **e**를 써넣어서 에러 위치로 이동할 수도 있고 **r**로써 에러를 무시하고 끝까지 시도하게 할 수 있다.

컴파일이 성공하면 pdf 미리보기 화면이 다음과 같이 나타난다.



2 자동 패키지 설치 기능에 대하여 알아두어야 할 사항

MiKTeX의 **mpm**은 매우 매력적이고 훌륭한 기능이지만 다음 사항을 알아두어야 당황하지 않을 수 있다.

- (1) MiKTeX에서 제공하는 패키지라 하더라도 스타일/매크로 패키지가 아니라 오픈타입 폰트 패키지인 경우에는 문서 내의 폰트 호출 방법에 따라 자동 설치가 동작하지 않을 수 있다. 다음 장의 16 페이지에 설명해두었다.
- (2) 일부 패키지의 경우 설치가 되었음에도 컴파일에 실패한 것처럼 표시되는 경우가 있다. 이 때는 컴파일을 다시 시도하면 된다.
- (3) 가끔 MiKTeX 미러 서버가 먹통이 되거나 서버 업데이트가 진행 중일 때 자동 패키지 설치가 동작하지 않을 수 있다. 이 경우는 어쩔 도리가 없으므로 문제가 해결될 때까지 기다려야 한다. 대체로 수 시간 정도가 지난 후에 다시 시도하면 성공한다.

참고 MiKTeX은 미러 서버를 자동으로 선택하는 것이 디폴트이지만 사용자가 이를 강제 지정할 수도 있다. 만약 위의 (3)과 같은 상황에서 긴급하다면 MiKTeX Console의 Updates에서 “Retrieve from”의 버튼을 클릭하여 (현재) 안정적으로 동작하는 서버를 지정하면 된다. 일부 경험자의 보고에 의하면 우리나라의 naver, kakao 서버에서 문제가 생기면 일본의 jaist 서버를 선택함으로써 해결할 경우가 많았다고 한다.

폰트 설치

Noto Korean 폰트를 설치하자.

1 어떤 Noto 폰트를 설치할 것인가?

Noto Korean 폰트는 종류가 서너 가지 있다.

1. <https://fonts.google.com>에서 Korean으로 검색해서 찾을 수 있는 폰트. 이것은 Noto Serif KR과 Noto Sans KR이라는 이름으로 활용할 수 있는 폰트이다.
2. <https://github.com/googlefonts/noto-cjk/releases>에서 받을 수 있는 폰트. 이것은 Noto Serif CJK KR 또는 Noto Sans CJK KR이라는 이름의 폰트이다.

이 가운데 2번 항목의 CJK 폰트를 설치하기로 한다. 이것도 여러 종류가 있는데, Language Specific OTFs Korean이라는 것을 권장한다. 각각의 주소는 2023년 2월 28일 현재 다음과 같다.

- Serif: https://github.com/googlefonts/noto-cjk/releases/download/Serif2.001/08_NotoSerifCJKkr.zip
- Sans: https://github.com/googlefonts/noto-cjk/releases/download/Sans2.004/07_NotoSansCJKkr.zip

다운로드 받은 zip 파일을 풀어두자.

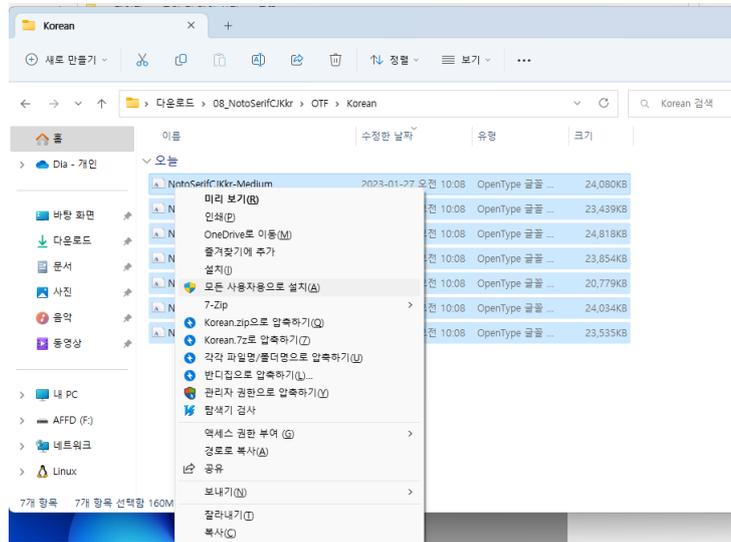
참고 KTUG 사이트와 같은 곳에서 볼 수 있는 한글 문서 예제는 Noto Serif KR/Noto Sans KR 폰트를 사용한 것이 현저히 많다. 만약 Noto CJK KR 폰트를 설치하였고 이런 예제에서 에러를 만나게 되었다면 폰트 이름을 자신이 설치한 것으로 찾아 고쳐서 시도하여야 한다.

1번의 Noto KR 폰트를 선호하는 사람도 많다. 폰트 크기가 상대적으로 작기 때문에 한자 관련한 문제가 심각하지 않다면 이것이 더 좋을 수 있다.

2 폰트 설치

Serif 폰트를 풀어둔 폴더를 열면 몇 개의 otf 폰트 파일이 있다. otf 파일을 전체 선택한 다음 **[Shift] + [Right Click]**하여 모든 사용자용으로 설치를 선택한다.

그냥 **[Right Click]** 후에 설치를 선택해도 설치는 가능하지만 여기서는 '모든 사용자용으로 설치' 하는 것으로 기술한다. 5장 5절을 보라.



3 설치 상태 점검

다음과 같은 파일을 컴파일하여 어려없이 진행되는지 확인한다.

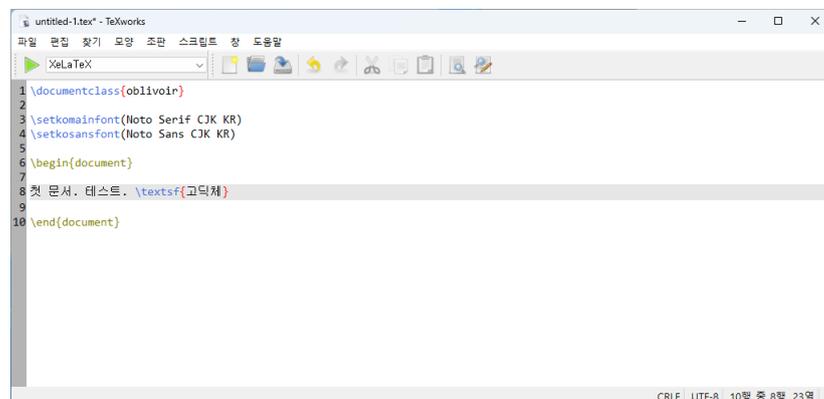
```
\documentclass{oblivoir}

\setkomainfont(Noto Serif CJK KR)
\setkosansfont(Noto Sans CJK KR)

\begin{document}

첫 문서. 테스트. \textsf{고딕체}

\end{document}
```



4 다른 폰트의 설치

트루타입 또는 오픈타입 폰트를 X_YTeX에서 활용하게 하려면 같은 방법으로 설치한다.

참고 자신이 선택한 폰트의 이름을 어떻게 확인하는가? 터미널을 열고

```
otfinfo -a <폰트파일이름>
```

이라는 명령을 내렸을 때 보여주는 이름을 쓰면 된다.

5 폰트 호출에 관한 주석

MiKTeX의 X_YTeX 또는 LuaTeX으로 트루타입이나 오픈타입 폰트를 활용함에 있어서 다음 사항을 알아두면 편리하다.

- (1) 폰트 이름으로 호출하는 경우는 `mpm`이 작동하지 않는다. 예컨대

```
\setmainfont{TeX Gyre Termes}
```

라는 코드는 만약 `tex-gyre` 폰트가 설치되어 있지 않다면 컴파일에 실패한다. 다만

```
\setmainfont{texgyretermes-regular.otf}
```

이와 같이 파일 이름으로 호출하면 `mpm`이 동작하며 자동 설치가 가능하다. 그리고 일단 설치된 이후에는 파일 이름으로든 폰트 이름으로든 다 동작한다.

- (2) `mpm`의 자동 설치가 동작하지 않으면 수동으로 설치할 수 있다. 다음과 같은 기본 `open-type`은 미리 설치해두는 것도 좋은 선택이다.

```
mpm --install noto
mpm --install tex-gyre
mpm --install tex-gyre-math
mpm --install xits
mpm --install stix2-otf
mpm --install libertine
```

각각 한 개씩의 패키지만 설치할 수 있으므로 위와 같이 호출한다.

참고 명령행을 이용하지 않고 MikTeX-Console을 실행하여 `Packages`에서 검색하여 설치할 수도 있다.

- (3) 외부 폰트를 설치함에 있어서 `Right-click` 후에 설치를 선택한 경우(즉, 자신만을 위하여 설치한 경우)에도 대부분의 폰트는 사용 가능하다. 파일 이름으로 호출하는 경우에는 어떠한 제약도 없고, 폰트 이름으로 호출하려 하는 경우 재시동이 필요할 수 있다.

만약 폰트 이름으로 호출이 잘 되지 않는 폰트가 있다면 “모든 사용자용으로 설치”하면 된다.

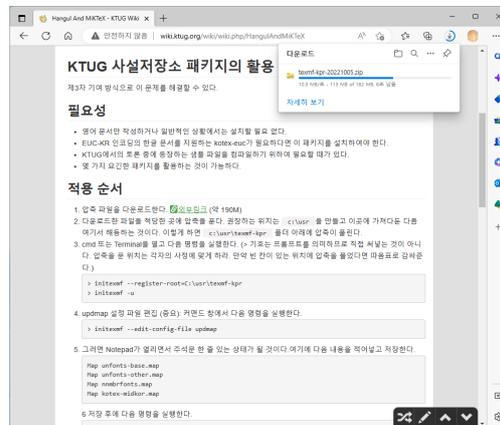
KTUG 사설저장소 패키지

KTUG 사설저장소에서 배포하는 패키지 중에 몇 가지는 한국어 문서를 작성하는 데 유용하다. 이 저장소는 $\text{T}_{\text{E}}\text{X}$ Live만을 지원하므로 $\text{M}_{\text{i}}\text{K}_{\text{T}}\text{E}_{\text{X}}$ 사용자에게 친절하지 않은데, 다음과 같은 절차를 거쳐 이를 $\text{M}_{\text{i}}\text{K}_{\text{T}}\text{E}_{\text{X}}$ 에서 사용할 수 있다.

이 절은 [KTUGWiki::한글과 MikTeX](#) 페이지에 실린 내용이다.

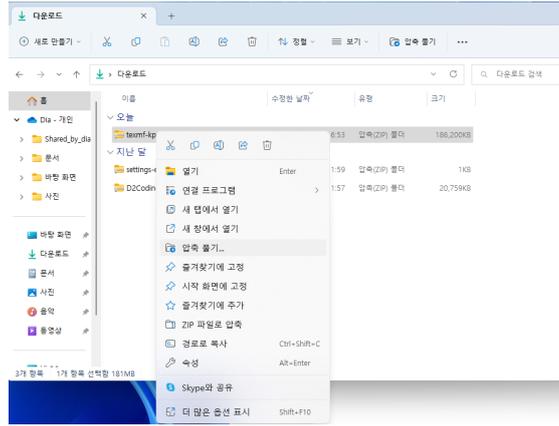
참고 일반적으로 한글 문서를 활용하는 데 KTUG 사설저장소가 필수적으로 필요한 것은 아니다. 그러므로 이 장은 건너뛰어도 좋다. 그러나 현재 진행하는 수업(또는 워크숍)의 일부 예제 문서를 컴파일하기 위해서 필요하므로 아래 “ $\text{X}_{\text{E}}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ 을 위해서 여기까지”라고 명시한 부분까지는 따라할 것을 권장한다.

1 설치

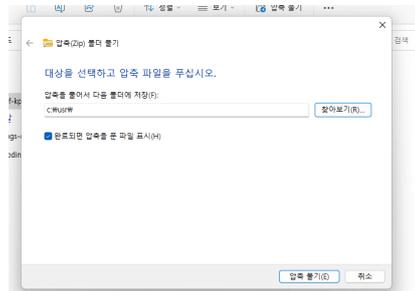


- **texmf-kpr.zip**(2023-02-28) 파일을 다운로드한다. 이 압축 파일을 `C:\usr\texmf-kpr` 폴더에 풀어놓아야 한다. 다음을 순서대로 따라하라.

– 다운로드받은 다운로드 폴더를 열어서, `texmf-kpr.zip`을 **Right Click**하여, “압축 풀기”를 선택한다.



- 압축을 풀 폴더를 보여주는 대화창이 열렸을 때, 다음 그림과 같이 c:\usr를 써넣는다.



- C:\usr\texmf-kpr이라는 폴더가 만들어지면서 압축이 풀리고 나면 C:\usr 위치로 탐색기 창이 열릴 것이다.

- 터미널을 열어서 다음 명령을 실행한다.

```
initexmf --register-root=C:\usr\texmf-kpr
initexmf -u
```

- 설치가 잘 되었는지 확인하기 위해 다음 명령의 결과가 jiwonlipsum.sty의 위치를 잘 가리키는지 본다.

```
kpsewhich jiwonlipsum.sty
```

2 폰트

C:\usr\texmf-kpr\fonts\truetype 폴더 아래에 hcr-lvt와 nanumttf라는 두 종류의 트루타입이 있다. 해당 폴더를 찾아 들어가서 폰트 파일 선택 후 우클릭하여 설치한다.

참고 Xe_{La}TeX을 위해서 여기까지. 이 이후 내용은 일종의 부록이다. 무시해도 좋다.

3 updmap 실행

위의 파일 묶음에는 현재 지원중지(obsolete) 상태인 kotex-euc 패키지가 포함되어 있다. 이 패키지는 EUC-KR 입력된 한글 문서를 pdflatex으로 처리하게 할 수 있으며, 아주 오래 전에 작성된 `\usepackage{hangul}`을 사용하는 문서를 컴파일할 수 있다.

또한 T_EXLive에 기본 탑재되어 있는 nanumtype1 폰트 이외에 unfonts type1을 kotex-utf에서 사용하게 할 수 있다. 이를 위하여 다음 절차를 거치면 된다.

- 다음 명령을 터미널에서 실행한다.

```
initexmf --edit-config-file updmap
```

- 그러면 Notepad가 열리고 주석문 한 줄이 있는 상태가 된다. 주석문은 건드리지 말고 그 뒤에 다음 내용을 그대로 써넣는다.

```
Map unfonts-base.map
Map unfonts-other.map
Map nnmbrfonts.map
Map kotex-midkor.map
```

- 저장 후에 Notepad를 닫고 다음 명령을 실행한다.

```
updmap
```

4 기타

이밖에 komkindex arara rule이 있는데 이것은 arara를 사용하지 않는 경우에는 해당사항이 없다. 만약 arara를 사용하고 있고 komkindex가 필요하다면 <http://wiki.ktug.org/wiki/wiki.php/HangulAndMiKTeX>의 해당 항목을 참고하여 설치하라. 이 안내서에서는 이에 대해 더 언급하지 않는다.

Visual Studio Code

1 왜 VS Code인가?

\LaTeX 을 본격적으로 쓰게 되면 대부분의 시간을 에디터와 함께 보내게 된다. 에디터가 얼마나 편리하고 손에 맞느냐에 따라 작업 생산성에 극명한 차이가 나기 때문에 자신에게 적합한 에디터를 발견하고 길들이는 것은 대단히 중요한 일이다.

Mi \TeX 이 기본 제공하는 \TeX works(9페이지)도 나쁘지 않은 에디터이고 \TeX Studio 라든가 \TeX maker 같은 \LaTeX 전용 에디터가 있으나, 이 안내서는 Visual Studio Code라는 범용 에디터를 \LaTeX 에디터로 사용하는 상황을 가정하였다. 이 에디터는 다른 전용 에디터와 달리 \LaTeX 사용 상황에 알맞게 약간의 설정 작업을 해주어야 한다. 이 일은 조금 귀찮지만 충분히 시도해볼 가치가 있다.

참고 만약 \TeX works나 \TeX Studio로 만족한다면 이 장은 건너뛰어도 좋다.

이 안내서는 VS Code에

- (1) [LaTeXworkshop](#)이라는 확장 프로그램을 설치하고
- (2) 한글 문서 작성 상황에 적합하도록 X_{\LaTeX} 을 기본 엔진으로 바꾸고
- (3) PDF 뷰어로 소스와 연동하는 것

까지를 설명하기로 한다. 더 많은 상세한 기능을 모두 설명하지 않는다.

2 VS Code를 주 에디터로 사용하기 위한 권장 설치 사항

이 안내서는 코딩 폰트를 D2Coding으로 하는 경우를 권장 사항으로 작성한다. 따라서 폰트를 미리 설치해 두어야 한다.

D2Coding Font 공개 코딩 글꼴이다. github.com/naver/d2codingfont에서 배포 zip 파일을 다운로드하여 풀면 D2Coding, D2CodingAll, D2CodingLigature라는 폴더가 생겨난다. 이 가운데 D2Coding에 있는 ttf 파일 두 개를 선택하여 설치하라.

참고 D2CodingLigature의 설치 여부, ttc로 설치할 것인지 ttf로 설치할 것인지는 모두 본인이 알아서 선택하면 된다. 잘 모르겠으면 위에서 제시한 대로 D2Coding ttf 2개를 선택하여 설치하면 되겠다.

3 VS Code의 설치와 **Shift** + **Ctrl** + **P**

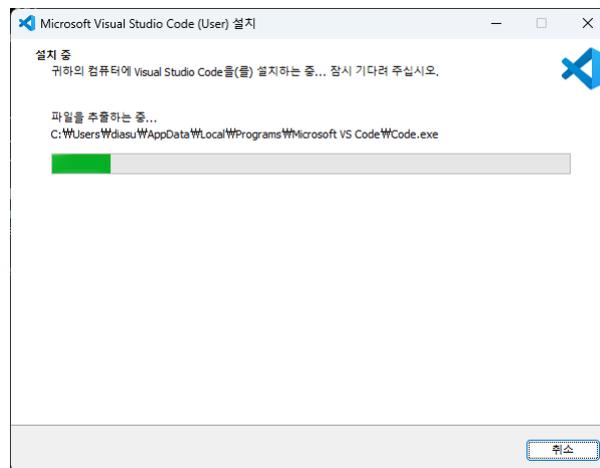
3.1 설치

winget으로 설치한다.

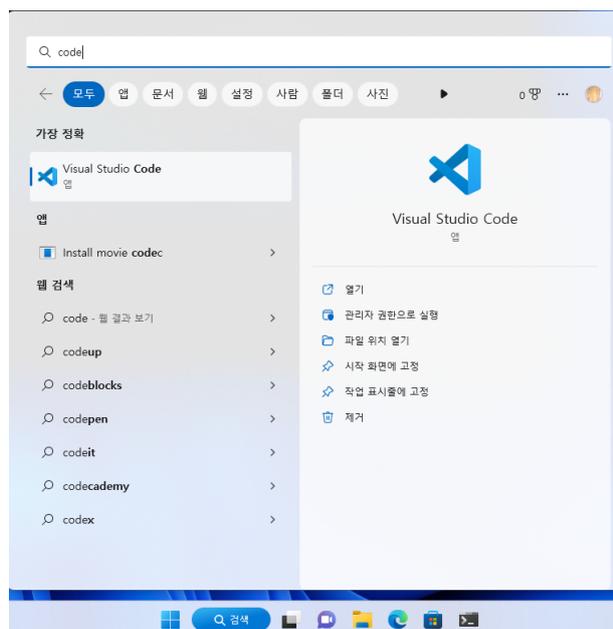
```
winget install Microsoft.VisualStudioCode
```

```
PS C:\Users\diasu> winget search VisualStudioCode
이름                장치 ID                버전  원본
-----
Microsoft Visual Studio Code Insiders Microsoft.VisualStudioCode.Insiders 1.75.0 winget
Microsoft Visual Studio Code           Microsoft.VisualStudioCode           1.74.3 winget
PS C:\Users\diasu>
PS C:\Users\diasu> winget install Microsoft.VisualStudioCode
```

그러면 VS Code의 설치가 시작된다.



설치가 완료된 후에 **Win** 키를 눌러서 `code`라고 입력하여 이 앱을 잘 실행할 수 있는지 확인하자.

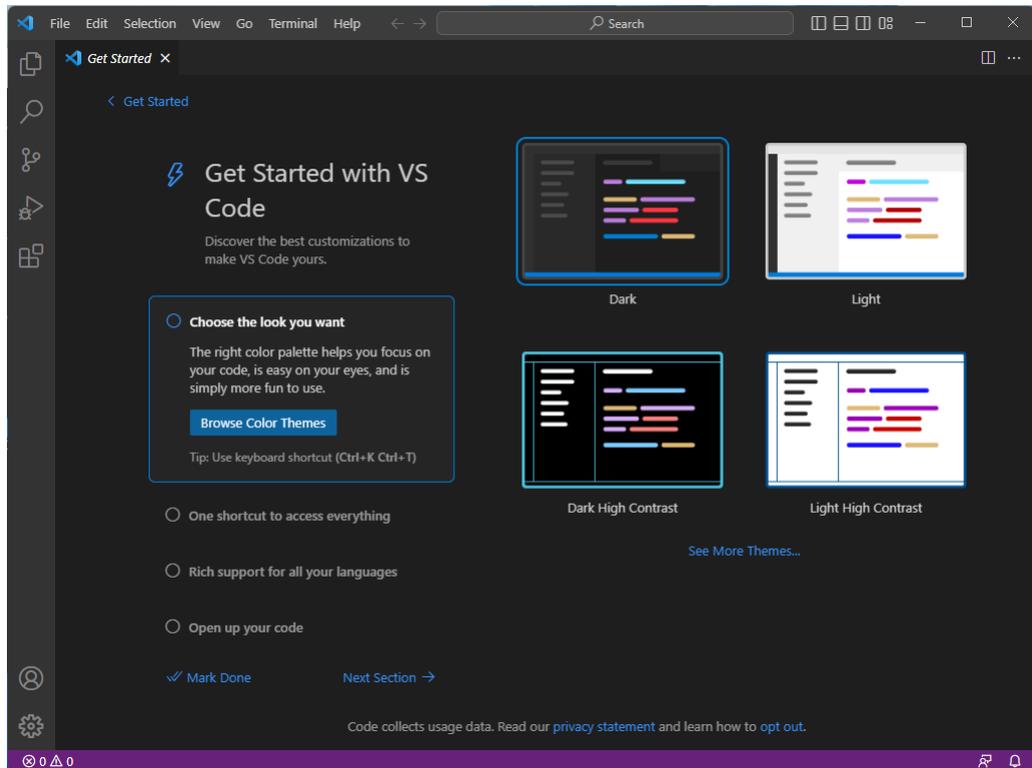


참고 VS Code는 Windows 탐색기의 Context Menu에 등록되기 때문에 편집할 수 있는 파일을 선택하고 **Right Click**하여 “연결 프로그램”을 찾아보면 VS Code로 해당 파일을 열 수 있다.

참고 `winget`에 의하지 않고 웹사이트에서 다운로드하여도 된다. <https://code.visualstudio.com/>을 방문하여 다운로드 버튼을 누른다. 이렇게 하는 경우에는 설치 위치와 권한을 사용자가 선택할 수 있는데, 이 안내서는 이러한 상황은 고려하지 않는다.

3.2 첫 실행

이제 VS Code를 처음 실행하면 다음과 같은 화면이 된다.



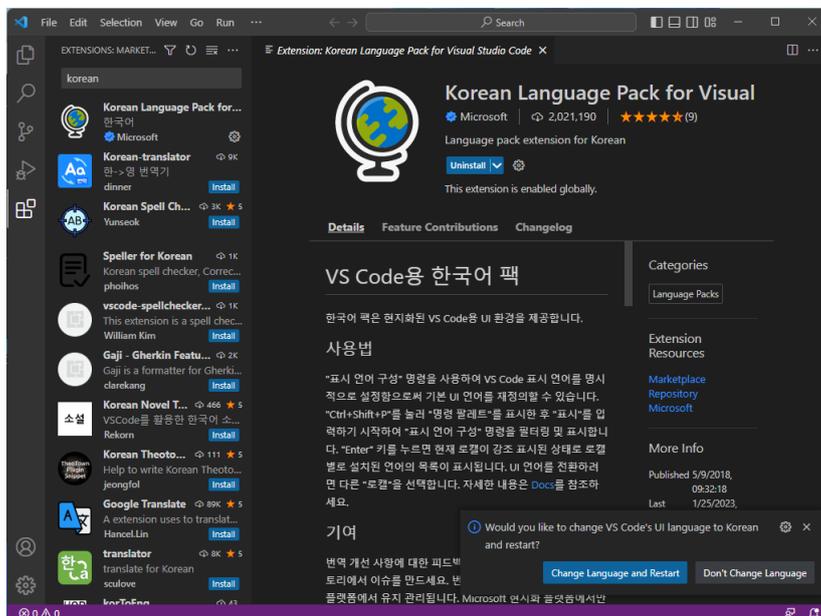
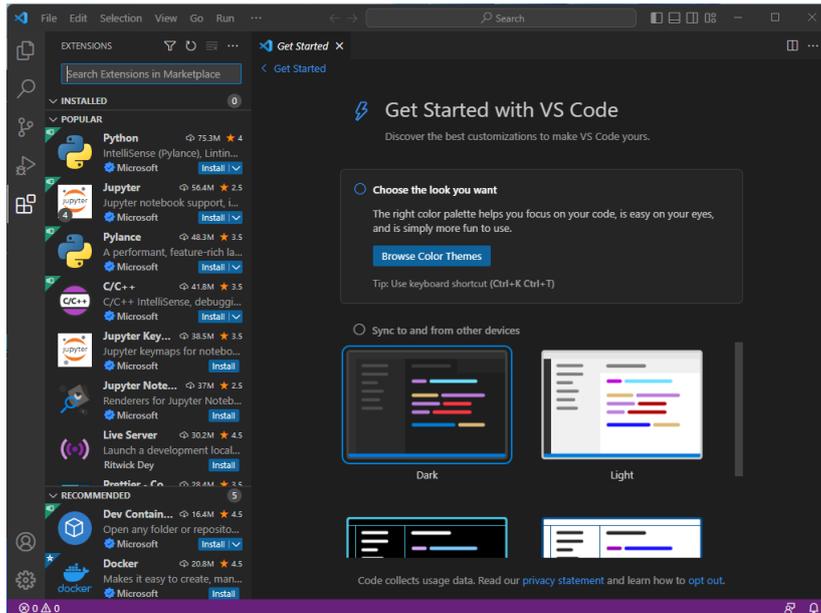
여기서 맨처음 해야 할 일은 VS Code의 인터페이스 언어를 한국어로 설정하는 것이다. 낯선(?) 영어로 고민하지 말고 한국어 언어 지원을 활성화하자.

이를 위해서 맨처음 발견해야 하는 것은 “Market Place”라는 곳을 열어보는 것인데, 화면 왼쪽에 있는  모양의 버튼을 누른다. 이곳에는 VS Code의 기능을 풍부하게 하는 수많은 확장 애드온(extensions)을 선택하여 활용할 수 있는 곳이다. VS Code의 최대 장점 중의 하나가 이 확장 기능이다.

다음 그림은 마켓플레이스를 열었을 때의 모양을 보여준다. 아무것도 설치한 것이 없으면 몇 가지를 추천하는데 일단 우리와는 큰 상관 없는 것이 많으므로 한국어 팩부터 설치하기로 한다.

제일 위에 있는 검색 바에 “korean”이라고 써넣으면 한국어 언어 패키지를 설치할 수 있다. 다음 그림과 같이 진행된다. 언어 패키지가 활성화된 이후에 VS Code를 한 번 닫았다가 다시 실행하여야 한다.

한국어화하지 않고 영어 그대로 쓰는 것이 더 멋져보일 수도 있다 (?).



참고 마켓 플레이스를 눌러서 선택하지 않고 VS Code의 “모든 명령 보이기(Show all commands)”로 명령 팔레트(Command Palette)를 연 상태에서 직접 명령을 지정하여 설치하는 방법도 있다. Configure Display Language 명령을 입력하고 “Korean”을 선택하면 된다. 물론, 다시 시작하여야 한다.

3.3 **Ctrl**+**Shift**+**P**

VS Code에는 명령 팔레트라고 부르는 조그마한 입력창이 존재한다. 이것은 에디터의 기능이 너무 많아서 메뉴로는 다 보여줄 수 없는 관계로 메뉴에 있거나 없거나 숨어 있는 기능까지 에디터 자체를 제어하는 모든 명령을 직접 입력하여 실행할 수 있는 곳이다.

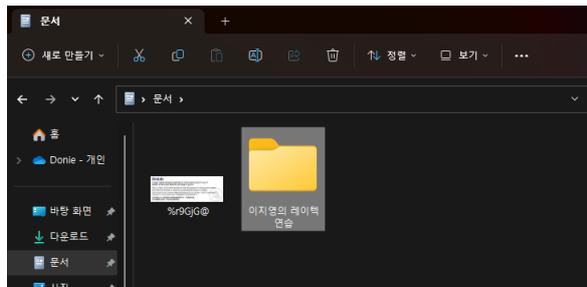
명령 팔레트를 여는 것은 **Ctrl**+**Shift**+**P** 단축키가 가장 쉽다. 이 단축키는 앞으로 자주 사용하게 될 것이므로 손에 익혀두자. 연습삼아 단축키를 누르고 “help”라고 써넣어 본다.

어떤 도움말을 준비하고 있는지 볼 수 있을 것이다.

3.4 폴더 열기와 작업 공간

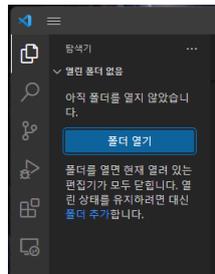
이제 파일을 하나 만들어서 저장해보려 한다.

먼저, 작업 폴더를 하나 만들자. 탐색기의 문서 폴더 아래에 “이지영의 레이텍 연습”이라는 폴더를 작성한다. (Right Click 후에 ‘새로 만들기’에서 ‘폴더’를 선택하여 새 폴더를 만들 수 있다.)



참고 MiKTeX의 장점 하나가 파일 이름에 한글, 띄어쓰기가 있어도 문서 처리에 문제가 발생하지 않는다는 점이다. 그러므로 폴더와 파일 이름을 자유롭게 지어도 된다. Windows용 TeXLive는 그렇지 않아서 반드시 ASCII 문자로만 띄어쓰기 없이 짓도록 권장하고 있다는 사실과 비교해보라. 다만 파일을 교환할 상대방이 TeXLive 사용자라면 파일 이름을 바꾸지 않는 한 컴파일에 곤란을 겪을 수 있다.

이제 VS Code에서 탐색기 버튼 — 왼쪽 바의 제일 위에 있는 아이콘 — 을 누르고 “폴더 열기”를 선택하여 방금 만든 폴더를 연다.



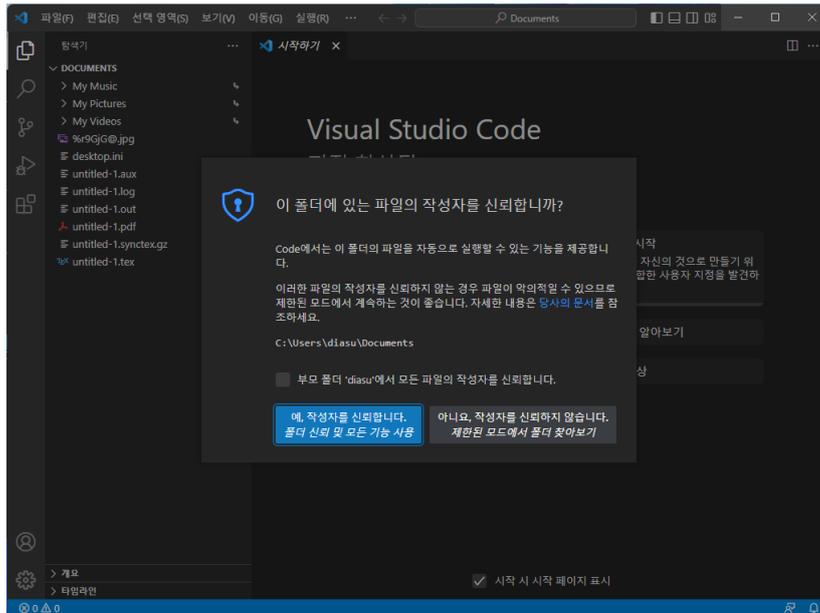
이 상태에서 새로 편집하는 파일은 방금 연 폴더에 저장된다. 이곳이 작업 공간이 되는 것이다.

적당한 내용으로 새로운 파일을 하나 작성 (Ctrl+N)하고 저장하여 본다. 주로 LaTeX 파일을 다루게 될 것이므로 확장자를 .tex으로 하는 것이 좋다.

이후, 이 아이콘을 눌러서 작업 중인 파일을 즉시 불러올 수 있다.

3.5 신뢰할 수 있는 작업 공간

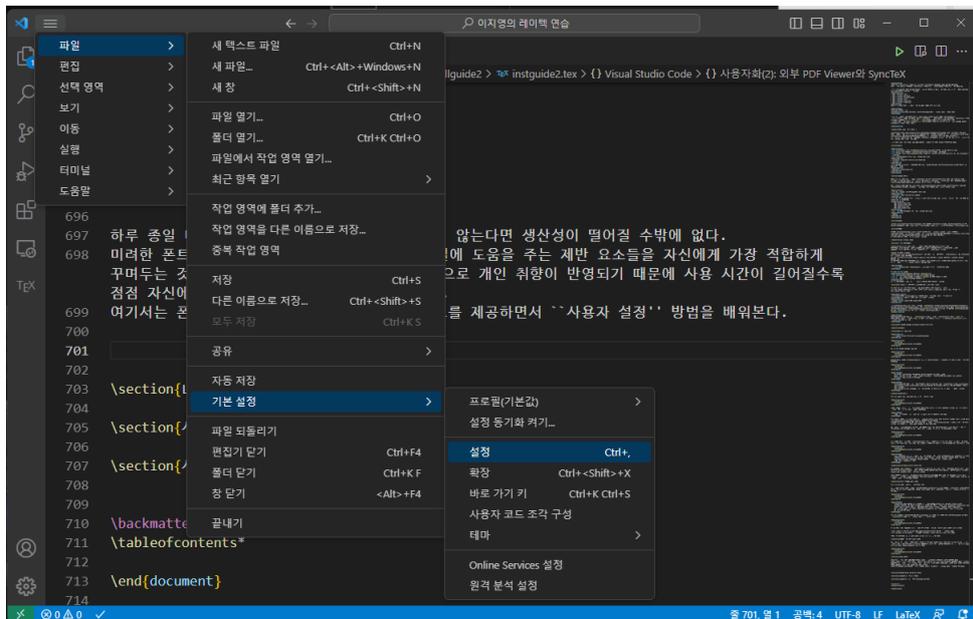
다음 그림과 같은 상황은 위에 설명한 방법으로 연 작업 공간이 아닌 곳에 있는 파일을 열었을 때 주로 만나게 된다. 한번 열어보는 파일이라면 신뢰하는 것으로 하고 작업을 계속하면 되지만 지속적으로 작성하는 문서라면 작업 공간을 설정하는 것이 더 낫다.



4 사용자 설정

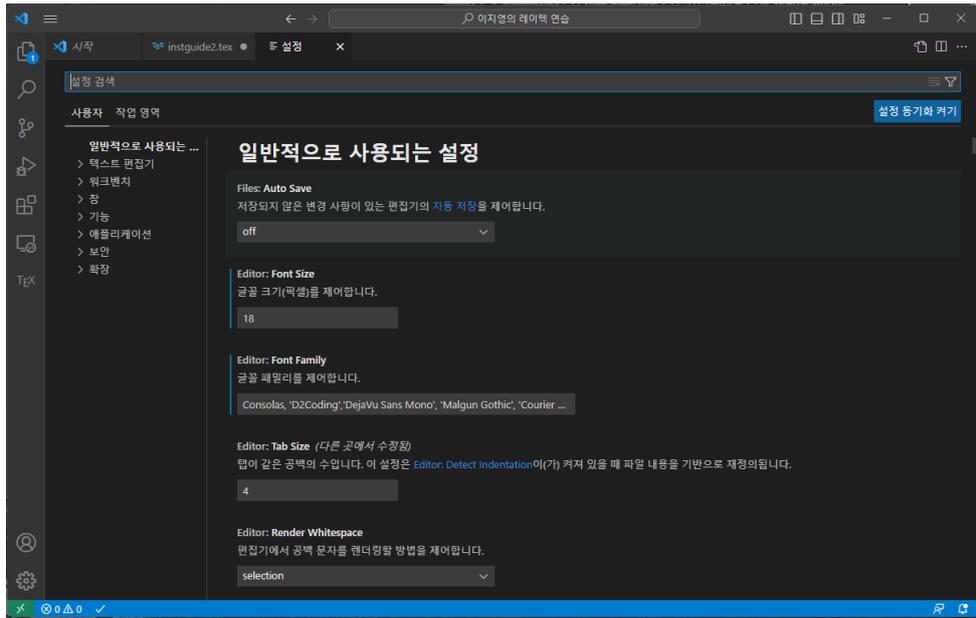
하루 종일 해야 하는 편집기가 편하게 느껴지지 않는다면 생산성이 떨어질 수밖에 없다. 미려한 폰트, 적절한 행간, 그리고 입력과 컴파일에 도움을 주는 제반 요소들을 자신에게 가장 적합하게 꾸며두는 것은 권장할 만한 일이다. 이것은 전적으로 개인 취향이 반영되기 때문에 사용 시간이 길어질수록 점점 자신에게 맞는 것을 찾아갈 수 있을 것이다. 여기서는 폰트와 색상 설정에 대해서 약간의 힌트를 제공하면서 “사용자 설정” 방법을 배워본다.

메뉴의 **파일 > 기본 설정** 아래의 하위 메뉴로 (1) 설정, (2) 바로가기 키, (3) 테마 등을 선택해둘 수 있는데, 우선 (1) 설정에 대해 간단히 알아본다.



설정에서는 폰트 크기와 폰트 종류 정도를 지정해본다. 다음 그림은 현재 필자의 설정이다.

즉, 폰트 크기를 18로 하고 폰트 종류(font family)를 Consolas와 D2Coding을 차례로



지정했다. 이렇게 해주면 한글은 D2Coding으로, 영문자는 Consolas로 표시한다. (폰트 크기를 18로 한 것은 필자의 시력 때문이므로 자신에게 맞게.)

이 상태에서 **Ctrl+Shift+P**를 눌러서 명령 팔레트를 열어 “user settings” 정도를 입력하면 “Preferences: Open User Settings (JSON)”이라는 명령과 “Preferences:Open User Settings”라는 명령이 사용 가능함을 보여줄 것이다. 이 가운데 (JSON)이 붙은 것은 사용자 설정을 직접 편집할 수 있게 하는 것이고, (JSON)이 없는 것은 방금 연 설정 창이다. (JSON)이 붙은 “기본 설정: 사용자 설정 열기(JSON)”을 선택하자. 그러면

```
"editor.fontSize": 18,
"editor.fontfamily": "'Consolas', 'D2Coding', 'DejaVu Sans Mono'"
```

이런 사항이 기록되어 있을 텐데 이것은 조금 전 ‘설정’ 창에서 행한 설정값이 적혀 있는 것이다.

앞으로 사용자 설정 파일을 직접 편집하는 방식으로 \LaTeX 관련 설정을 하는 방법을 소개할 것임을 알아두자.

이 파일은 각 항목이 "key": "value" 형식으로 되어 있고 한 항목이 끝나면 쉼표(,)를 적어야 한다.

5 LaTeXWorkshop Extension 설치

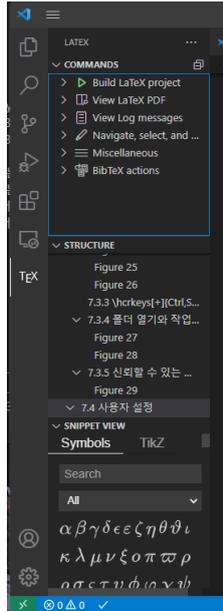
이 확장은 James Yu가 만들고 관리하는 것으로 매우 풍부하고 강력한 \LaTeX 사용 환경을 제공하는 것으로 유명하다. VS Code로 .tex 파일을 열었을 때 이 확장을 추천해주기도 한다.

LaTeXWorkshop을 설치하자.

“마켓플레이스”(📦)를 열고 latexworkshop을 검색하면 바로 설치할 수 있다.

참고 **Ctrl+Shift+P**를 누르고 ext install latex-workshop을 입력하여 설치하는 방법도 있다.

.tex 파일을 열어보면 문법 강조가 활성화되는 것을 볼 수 있는데 이것은 latexworkshop의 기본 기능 가운데 하나이다. .tex 파일을 열어서 latexworkshop이 활성화되면 **Ctrl+Alt+X** 단축키나 왼쪽 아이콘바의 **TeX** 아이콘을 눌러서 latexworkshop의 메뉴를 열어볼 수 있다. 이것을 훑어보면 어떤 일을 할 수 있는지 대략 알 수 있게 된다.



참고 현재까지 진행한 상태에서 tex 파일을 저장하면 latexworkshop의 기본 설정에 따라서 latexmk/pdflatex이 동작하면서 build를 시도한다. 그러나 만약 소스가 Xe_{La}TeX을 위한 것이라면 폰트 활용 관련 이런저런 에러가 발생했다면서 귀찮은 메시지가 나타날 것이다. 아래 이어지는 설정을 적용하면 이 문제는 사라지므로 일단 무시한다. 혹은, 아예 Perl이 동작하지 않는다는 메시지를 만날 수도 있는데 이것은 Perl 설치를 미루었기 때문이다. VS Code가 채택하고 있는 기본 빌드 유틸리티가 Perl 스크립트인 latexmk이기 때문이다.

6 사용자화: 컴파일

이어지는 ‘사용자화’ 관련 내용은 개인에 따라 다른 선택을 할 수 있다. 이 안내서의 목적은 일단 따라해보는 것이므로, VS Code에 익숙한 경우라면 무시한다.

6.1 저장시 빌드

저장시 빌드는 긴 글을 쓰는 사람에게 꽤 편리한 기능이다. 그러나 LaTeX을 처음 배우거나 LaTeX 프로그래밍을 할 때는 오히려 방해가 된다.

latexworkshop을 처음 설치한 상태에서 (1) 기본 컴파일 엔진은 pdflatex, (2) 저장시 빌드(컴파일)은 켜짐 상태이다. 이 가운데 “저장시 빌드” 옵션을 끄도록 하자. 빌드(pdf 만들기)는 내가 원할 때 단축키를 눌러서 하는 방식으로 설정하겠다.

[Ctrl]+[Shift]+[P]를 눌러서 “사용자 설정 열기(JSON)”을 선택하여 settings.json을 연다. 여기서 다음 한 줄을 파일 마지막의 닫는 } 앞에 복사해 넣고 저장한다.

```
"latex-workshop.latex.autoBuild.run": "never",
```

직전 항목의 마지막에 쉼표가 있어야 한다.

참고 이 동작은 메뉴의 설정을 눌러서 열리는 일반 “사용자 설정”에서 autobuild로 검색하여 찾은 다음 선택할 수도 있다.

6.2 컴파일 엔진

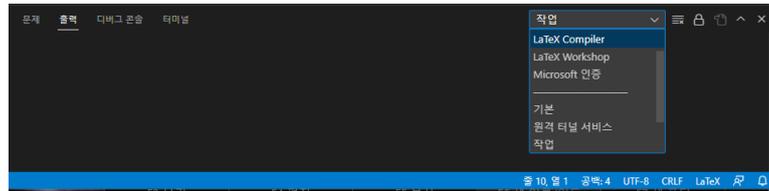
참고 만약 LaTeX 을 쓰는 목적이 한글이 아니고 영어로 된 글이라면 pdfLaTeX 은 좋은 선택이다. 굳이 XeLaTeX 으로 바꿀 필요 없는 것이다.

같은 방법으로 `settings.json`을 열어서, 다음 한 줄을 추가한다.

```
"latex-workshop.latex.recipe.default": "latexmk (xelatex)",
```

참고 VS Code에서 컴파일(빌드) 엔진을 선택하는 방법은 (1) 항상 특정 엔진으로 컴파일되게 하는 것, (2) 마지막에 사용한 레시피를 다시 사용하게 하는 것, (3) 파일의 매직 코멘트를 읽어서 선택하게 하는 것 등 여러 방법이 있다. 나중에 익숙해지면 이 중 자신에게 맞는 것을 골라서 설정해보는 것을 권한다.

이제 `Ctrl+Alt+B`를 눌러서 현재 편집 중인 파일을 컴파일해보자. 컴파일 과정을 보려면 `Ctrl+Shift+U`를 누른 다음 “작업”을 “LaTeX Compiler”로 하면 된다.



참고 편집 중인 화면에서 컴파일러를 실행하는 방법은 여러 가지가 있는데, 단축키 이외에도 `Ctrl+Shift+P`로 명령 팔레트를 열어서 “recipe” 정도를 써넣으면 “LaTeXworkshop: Build with Recipes” 항목을 선택할 수 있고 이로부터 원하는 컴파일러를 동작하게 할 수 있다. 이것은 `Ctrl+Alt+X`로 `latexworkshop` 메뉴를 열어서 “Build Latex Project”에서 레시피를 고르는 것과 마찬가지로이다.

7 PDF Viewer와 SyncTeX

7.1 PDF를 여는 방법

컴파일 과정에서 에러가 발생하고 이를 발견하고 수정하는 것은 이 안내서의 범위를 넘어선다. 어떻게든 소스를 에러 없이 작성하여 컴파일까지 성공해보자.

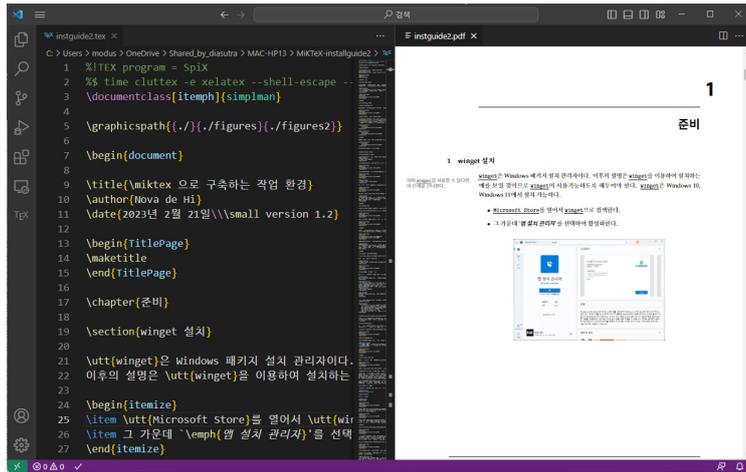
이제 앞서 작성하던 파일을 하나 열어보자. 그런 다음 `latexmk (xelatex)` 레시피로 컴파일한다. 에러 없이 컴파일되었다고 가정하겠다.

VS Code에는 내장 PDF Viewer가 있다. `Ctrl+Alt+V` 단축키를 누르면 (PDF가 잘 생성되었을 때) PDF 파일이 화면 오른쪽에 열린다.

참고 `latexworkshop` 메뉴에서 “View LaTeX PDF”의 “View in VS Code tab”이라는 명령으로 바로 열어볼 수 있다. 물론 명령 팔레트에서 “VSCode tab”으로 검색하는 편이 빠르다.

이 때문에 긴 글을 쓰는 사람에게 “저장시 빌드”가 편리할 때가 있다고 한 것이다. 그러나 무수한 에러와 싸워야 하는 처음 사용자에게는 저장시 빌드가 오히려 방해가 되는 것이어서 이 안내서는 그것을 끄도록 안내했다.

우리는 앞서 “저장시 빌드” 옵션을 꺼두었기 때문에 왼쪽 소스 편집창의 내용이 변해도 PDF의 변경을 바로 알아차릴 수 없지만 만약 “저장시 빌드”가 켜져 있다면 소스를 변경하고 저장하는 순간 컴파일이 이루어지고 PDF의 내용도 변화한다.



PDF 보기 창의 적당한 위치에서 **Ctrl**+**Click**하면 그 위치에 해당하는 곳으로 커서가 이동한다. 그리고 편집창에서 **Ctrl**+**Alt**+**J**를 누르면 현재 커서가 있는 곳에 해당하는 위치로 PDF 보기 위치도 바뀐다. 주의할 것은, 오른쪽 뷰어 창이 열리지 않은 상태에서는 **Ctrl**+**Alt**+**J** 키가 동작하지 않는다는 것이다. 그러므로 먼저 **Ctrl**+**Alt**+**V**로 PDF 뷰어를 열고, 그 다음에 “소스 위치에 해당하는 위치로 PDF 열기”(Forward Search)를 해야 한다.

이와 같이 소스 편집창과 PDF 보기 창을 상호 동조하는 것을 SyncTeX이라고 하는데, 흔히 소스에서 PDF로 찾아가는 것을 “Forward Search”, PDF 창에서 해당하는 소스 위치로 이동하는 것을 “Inverse Search” 또는 “Backward Search”라고 부른다. 이 기능이 잘 동작하는지 확인해두어야 한다.

참고 이 안내서 작성자의 개인적인 취향 때문에 처음에는 외장 뷰어를 설치하고 설정하는 내용을 포함하고 있었다. 그러나 그 과정이 너무 번거로워서 처음 사용자를 당황하게 만든다는 지적이 있어서 이를 부록으로 돌렸다. 관심있는 분은 부록 31 페이지 SumatraPDF 관련 내용을 참고하기 바란다.

lshort-ko를 읽자

이제 \LaTeX 을 공부할 준비가 갖추어졌다. 당연히 lshort-ko라는 문서를 읽어보아야 한다. 터미널을 열고, 다음 명령을 써넣는다.

```
texdoc lshort-ko
```

처음에 이 문서를 Mi \TeX 이 다운로드할 것이다. 원하는 문서가 열렸는가? 『 $\text{\LaTeX}2_{\epsilon}$ 입문 — 143분 동안 익히는 $\text{\LaTeX}2_{\epsilon}$ 』라는 제목의 문서인데, 이제 차근차근 이 문서의 내용을 따라하면서 공부하면 된다. 이 책을 읽고나면 ‘ \LaTeX ’을 ‘라텍스’라고 읽으면 안된다는 것을 알게 된다.

VS Code와 LaTeXworkshop은 너무 강력한 도구라서 모든 기능을 전부 설명하는 것이 거의 불가능하다. 그것은 \LaTeX 이 너무 강력한 도구라서 그것으로 무엇을 할 수 있는지 설명하는 것이 불가능한 것과 거의 같다.

기타 유틸리티

A Python과 pygments

이 안내서의 상황과 관련하여 Python이 필요한 이유는 주로 `pygmentize`라는 유틸리티 때문이다. `minted` 패키지로 소스코드리스팅을 하는 경우에 이 유틸리티의 도움이 필요하다. 예를 들면 이 안내서가 그렇게 작성되었다.

Python도 터미널에서 `winget`으로 설치할 수 있다. 다른 목적으로 Python을 이용하는 경우가 아니고 오로지 이 안내서의 상황만을 고려한다면, 다음과 같은 명령을 내리면 된다.

```
winget install Python.Python.3.12
```

설치 후에 터미널을 한 번 닫았다가 열어준다. 그리고 다음 명령을 다시 내린다.

```
pip install pygments
```

참고 만약 이 단계에서 `pip`라는 명령이 없다고 하면서 실행되지 않으면 Python이 설치된 폴더를 `PATH` 환경 변수에 추가하여야 하는데, 이 문제는 일반적으로 `winget`으로 설치한 경우에 발생하지 않을 것이다.

참고 `minted` 패키지가 사용된 문서는 `-shell-escape`라는 옵션을 붙여서 컴파일하여야 한다.

B VS Code의 외부 뷰어

VS Code의 PDF Viewer를 (내장 뷰어가 아니라) SumatraPDF라는 외부 유틸리티로 설정하려면 다음과 같이 한다.

B.1 SumatraPDF 설치

터미널에서 다음 명령을 내린다.

```
winget install SumatraPDF.SumatraPDF
```

참고 SumatraPDF는 **웹페이지**로부터 다운로드하여 설치할 수 있다. 이 경우에는 설치 위치와 권한을 지정할 수 있는데, 이 안내서는 그렇게 설치된 경우를 고려하지 않는다.

B.2 외부 뷰어 설정

내장 뷰어는 화면이 좁은 경우에 에디터의 일부를 차지하게 되어 답답한 느낌이 들 때가 있고 그외에도 몇 가지 불만스러운 점이 있어서 SumatraPDF라는 pdf 뷰어를 설치하고 이를 외부 뷰어로 활용하는 것이 좋을 때가 있다.

그러므로, 만약 외장 뷰어가 자신이 원하는 바가 아니고 내장 뷰어로 충분하다면 이 절의 내용은 무시해도 좋으며, 혹시 이 설정이 뜻대로 되지 않는다고 해도 그냥 내장 뷰어를 쓰면 되는 것이다.

이를 위하여, 한 가지 확인하여야 할 것이 있다. 그것은 **자신의 이름**이다. 터미널을 열었을 때 프롬프트에 표시되는 자신의 이름을 확인하자. 예를 들어 프롬프트가 대략 다음과 같다면

```
C:\Users\sungguri>
```

자신의 이름은 sungguri인 것이다.

참고 터미널 명령행에서 `echo %userprofile%` 명령을 내리면 자신의 홈 폴더 위치를 출력한다. 이것으로 자신의 홈 폴더와 '이름'을 확인할 수 있다.

`Ctrl+Shift+P`로 명령 팔레트를 연 다음 “사용자 설정 열기(JSON)”을 선택하여 설정 편집 상태로 들어간다. 여기에 아래 내용을 복사하여 넣자.

```
"latex-workshop.view.pdf.viewer": "external",
  "latex-workshop.view.pdf.external.synctex.command": %
"C:/Users/sungguri/AppData/Local/SumatraPDF/SumatraPDF.exe",
  "latex-workshop.view.pdf.external.synctex.args": [
    "-forward-search",
    "%TEX%",
    "%LINE%",
    "-reuse-instance",
    "-inverse-search",
    "code.cmd -r -g \"%f:%l\"",
    "%PDF%"
  ],
  "latex-workshop.view.pdf.external.viewer.command": %
"C:/Users/sungguri/AppData/Local/SumatraPDF/SumatraPDF.exe",
```

위의 코드를 복사할 때 행의 끝이 %로 끝난 두 곳은 다음 줄과 행이 잘라지지 않는다는 의미이다. 이 안내서의 공간이 좁아서 한 줄을 두 줄로 나누어 표시한 것이므로 주의한다.

그런 다음 위의 `C:/Users/sungguri` 부분을 자신의 '이름'으로 바꾸어 넣는다. 예컨대 이름이 chunhyang이라면

```
C:/Users/chunhyang/AppData/....
```

와 같은 모양이 되게 해야 한다. 이렇게 고쳐야 할 곳이 두 곳 있다.

설정 저장 후 VS Code를 재실행한다.

B.3 SyncTeX

SyncTeX이란 편집기와 PDF Viewer 사이에서 원하는 위치로 이동하는 것이다. 이 상태에서 `Ctrl+Alt+V`를 누르면 SumatraPDF가 열려야 한다.

SumatraPDF의 특정한 위치에서 double click해보자. 그러면 (잠깐 검은 화면이 뜨면서) 편집기 상의 커서 위치가 PDF에서 클릭한 위치에 해당하는 곳으로 이동하여야 한다.

이번에는 에디터의 특정한 행(예를 들면 800행)에 커서를 두고 `Ctrl+Alt+J`를 눌러보자. 열려 있는 SumatraPDF에서 화면 이동이 일어나면서 잠깐 파랗게 위치를 보여줄 것이다.

참고: 기존에 SumatraPDF가 열려 있지 않은 것이 좋다.

C arara와 Java Runtime

arara 유틸리티를 사용하려면 Java Runtime이 설치하여야 한다. 이 안내서는 이에 대하여 더 상세히 다루지 않는다.

차례

제1장 준비	1
1 winget 설치	1
2 터미널 열기	2
제2장 MiKTeX 설치	4
1 winget으로 설치	4
2 설치 후	6
3 Perl 설치(거의 필수)	8
4 기타 유틸리티(선택)	8
제3장 TeXworks 설정	9
1 Preferences	9
제4장 첫 문서 처리 및 자동 패키지 설치	11
1 첫 문서 작성	11
2 자동 패키지 설치 기능에 대하여 알아두어야 할 사항	12
제5장 폰트 설치	14
1 어떤 Noto 폰트를 설치할 것인가?	14
2 폰트 설치	15
3 설치 상태 점검	15
4 다른 폰트의 설치	16
5 폰트 호출에 관한 주석	16
제6장 KTUG 사설저장소 패키지	17
1 설치	17
2 폰트	18
3 updmap 실행	19
4 기타	19
제7장 Visual Studio Code	20
1 왜 VS Code인가?	20

2 VS Code를 주 에디터로 사용하기 위한 권장 설치 사항	20
3 VS Code의 설치와 <code>[Shift]+[Ctrl]+P</code>	21
4 사용자 설정	25
5 LaTeXWorkshop Extension 설치	26
6 사용자화: 컴파일	27
7 PDF Viewer와 SyncTeX	28
제 8 장 lshort-ko를 읽자	30
부록 기타 유틸리티	31
A Python과 pygments	31
B VS Code의 외부 뷰어	31
C arara와 Java Runtime	33