

# 초보자를 위한 ko.T<sub>E</sub>X

Joohee Jeong

2011 년 4 월 28 일

## 차 례

1	T <sub>E</sub> X 을 공부하려면	1
2	설치, 그리고 첫 사용	2
3	텍과 TeXworks 에 대하여	5
4	기본 사용법	6
4.1	문서클래스와 패키지 . . . . .	6
4.2	코멘트, 수식, 특수문자, 빈칸, 줄바꿈, 명령 . . . . .	7
4.3	타이틀 페이지, 목차 . . . . .	10
4.4	그래픽 이미지 . . . . .	11
5	프리젠테이션 슬라이드	12
5.1	포함 파일과 테마 . . . . .	12
5.2	타이틀 페이지 . . . . .	14
5.3	목차, 프레임과 슬라이드, 오버레이 . . . . .	16

## 1 T<sub>E</sub>X 을 공부하려면

이 문서는 텍 (T<sub>E</sub>X) 을 처음으로 접하는 사람을 위한 사용설명서이다. 이것만 읽고도 나름 괜찮은 문서와 프리젠테이션 슬라이드를 만들 수 있을 것이나, 텍에 대한 지식을 충분히 갖추고 싶다면 아래의 문서들을 읽어 보기를 권한다.

- (1) [The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>](#) by Tobias Oetiker, Version 5.01, Apr.06, 2011

- (2) [L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 입문](#) by Tobias Oetiker, 김강수, 이기황, MIKA, 김지운, 샘처럼 옮김 Version 4.17.kr, Nov.05, 2005
- (3) [한국어 텍 ko.T<sub>E</sub>X v0.1.0 사용 설명서](#) by 은광희, 김도현, 김강수, 2007년 7월
- (4) [The BEAMER class User Guide for version 3.10.](#) by Till Tantau, Joseph Wright and Vedran Miletic, Jul.12, 2010

이상 4개의 문서는 모두 무료이며 제목을 클릭하면 PDF 파일을 다운 받을 수 있다. 위의 문서 중 하나만 읽고 말 거라면 (2)를 권한다. 이것은 (1)의 번역본이며 최신 버전이 아닌 것이 흠이다. (3)은 한글 텍에 대한 세부 사항을 개발자들이 직접 설명한 것이다. (4)는 프리젠테이션 슬라이드 작성을 위한 문서인데, 양이 조금 많다.

좀 더 전문적인 지식을 얻기를 원하는 분에게는 아래의 책들을 추천한다.

- (1) Donald E. Knuth. *The T<sub>E</sub>Xbook, Volume A of Computers and Typesetting.* Addison-Wesley (1986)
- (2) Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X: A Document Preparation System.* Addison-Wesley, (1994)
- (3) Helmut Kopka et. al. *Guide to L<sup>A</sup>T<sub>E</sub>X* (4th edition). Addison-Wesley (2003)
- (4) Frank Mittelbach et. al. *The L<sup>A</sup>T<sub>E</sub>X Companion* (2nd edition). Addison-Wesley (2004)
- (5) Michel Goossens et. al. *The L<sup>A</sup>T<sub>E</sub>X Graphics Companion* (2nd edition). Addison-Wesley (2007)

위의 책 중 하나만 읽고 싶다면 (3)을 권한다.

그리고 마지막으로 이 강의노트는 [초보자를 위한 ko.T<sub>E</sub>X](#)에서 언제든지 다운 받을 수 있다.

## 2 설치, 그리고 첫 사용

ko.T<sub>E</sub>X Live 2010을 Windows Vista/7에 설치하는 과정을 설명하겠다.

<http://www.ktug.or.kr>에서 kotexlive2010-full.exe를 다운 받은 후 아이콘을 더블클릭하면 설치된다. 설치파일의 크기는 1.6G 정도이다. 다운과 설치에 시간이 수십 분 소요될 것이다. 컴퓨터의 사양이 낮을 경우 설치에 시간이 오래 걸려 혹시 중단된 것이 아닌가 의심이 갈 수 있으나 계속 기다리면 결국 설치가 완료될 것이다.

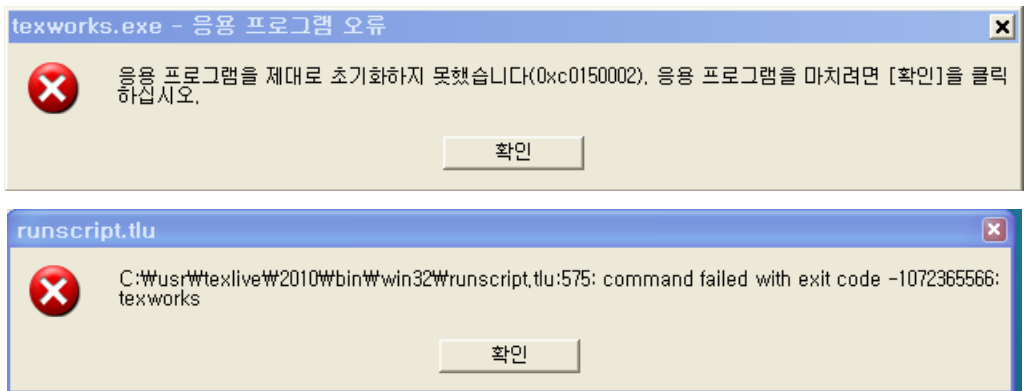
설치 디렉토리는 디폴트에서 다른 것으로 변경하면 제대로 되지 않는 경우가 있다고 하니 디폴트 디렉토리를 그대로 사용하도록 한다.

설치 도중에 ‘글꼴 캐시 정보를 지금 생성할까요?’라고 묻는 메시지 상자가 표시되면 ‘예’를 선택한다. 그러면 커맨드 창이 뜨고 컴퓨터가 정지된 것처럼 보이지만, 이게 정상이니 그냥 가만히 기다리면 1,2분 이내로 커맨드 창에 글씨들이 마구 뜨면서 설치가 잘 진행될 것이다.

아래의 왼편 그림은 설치 파일 아이콘이고 오른편 그림은 설치를 마치면 바탕화면에 생성되는 TeXworks의 아이콘이다.



설치가 제대로 되지 않으면 TeXworks를 실행시켰을 때 다음과 같은 에러 메시지가 뜨는 수가 있다.



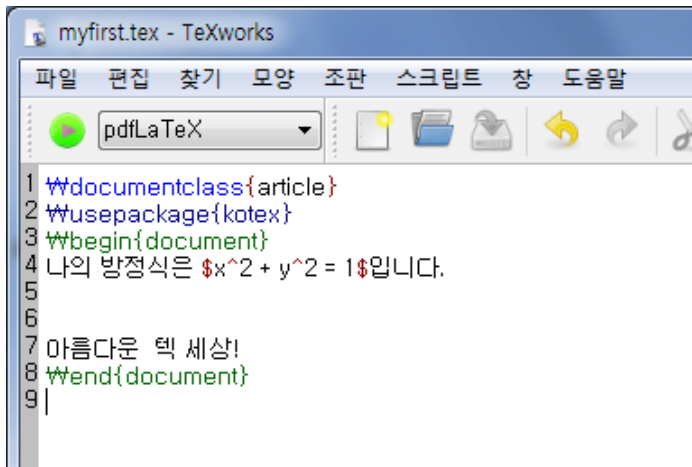
Windows XP 시스템에 설치하면 이런 문제가 발생하는 경우가 있다고 하며 뽕죽한 해결책은 알려져 있지 않다. 프로그램을 완전히 삭제하고 재설치 하면 제대로 설치되는 수가 있다고 한다. 그런데 프로그램 그룹의 Uninstall 이나, 제어판 - 프로그램에서 kotex Live를 찾아 제거하면 완전히 삭제되지 않는 것 같으니 탐색기에서 프로그램 관련 디렉토리를 찾아서 수동으로 삭제하는 것이 좋을 것이다. C:\usr과 C:\삭용 작\username\.texlive2010의 두 개다. 그리고 패키지파일인 tluupdateall.bat는 초기설치 문제를 해결해 주는 것이 아니라 나중에 업데이트 할 때 잘 안 되는 문제가 발생하면 해결해 주는 것이라고 ktug 게시판에 나와 있다.

설치된 프로그램들은 시작 버튼의 ko.TeX Live 2010 프로그램 그룹에 가면 나온다. 기본 작업환경을 제공하는 프로그램은 TeXworks이고 패키지 관리자는 TeX

Live Manager 이다. 그리고 또 TeXdoc GUI 라는 것이 있는데 이걸 매뉴얼들을 쉽게 찾아 볼 수 있는 유틸리티이며 제법 유용하다.

만일 기존에 MiKTeX 등 다른 텍 배포판(TeX distribution)을 사용했다면 이를 제거하지 않고 그냥 설치해도 된다. 다만 이렇게 하면 기존의 텍 프로그램이 제대로 작동하지 않게 될 것이다.

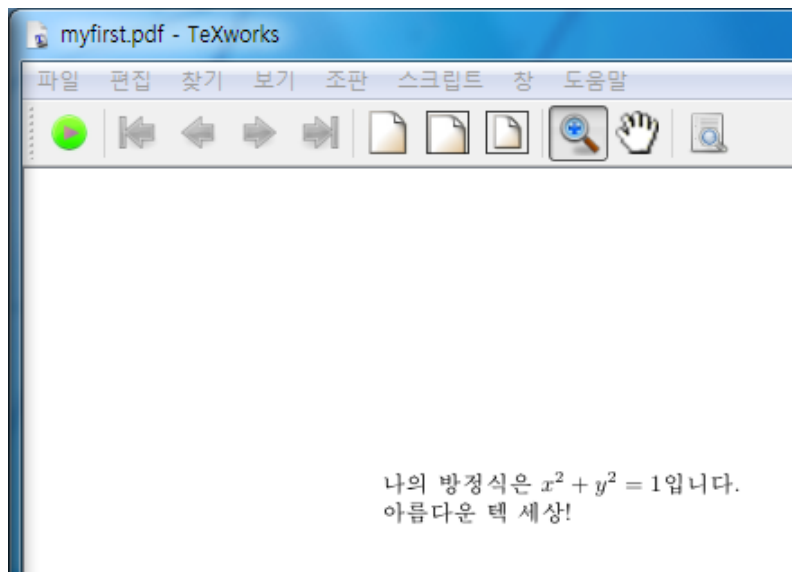
설치 후 바탕화면에 생성된 TeXworks 아이콘을 더블 클릭한다. 이때 나타난 창을 소스 에디터 창(*source editor window*)이라고 부른다. 나타난 그리고 아래 그림과 같이 문서를 작성한 후 화면 왼쪽 위의 초록색 삼각형을 클릭하여 컴파일(*compile*)한다. 이때 파일 이름을 요구하면 `myfirst.tex` 등 적당한 이름을 주면 된다. 디렉토리는 바탕화면 등 적당한 곳을 선택한다.



`myfirst.tex`을 소스 파일, 혹은 소스 텍 파일이라고 부르며 이것은 텍스트 파일이므로 notepad 등 다른 텍스트 에디터로도 읽거나 수정할 수 있다. 소스 파일을 (초록색 삼각형을 클릭하여) 컴파일 하면 출력(output) 파일로 `myfirst.pdf`가 생성되며, 이 외에도 많은 보조 파일들이 생성된다. 이들은 파일이름은 `myfirst`로 같지만 확장자가 `.aux`, `.log` 등으로 서로 다르다.

윈도즈 탐색기에서 확장자가 보이도록 설정해 놓지 않았다면 다음과 같은 방법으로 설정한다. XP에서는 [도구 - 폴더옵션 - 보기]에 가서 “알려진 파일 형식의 파일 확장명 숨기기”가 체크되어 있는 것을 해제하고 [확인]한다. Vista와 Win 7에서는 [구성 - 폴더 및 검색 옵션 - 보기]에 가서 마찬가지로 한다. 확장자가 탐색기에서 보이도록 설정하는 것은 텍 사용에 꼭 필요하지는 않으나 텍을 잘 이해하기 위해서 도움이 되니 반드시 이렇게 해 두기를 권한다.

그러면 또 하나의 PDF 뷰어 창(*viewer window*)이 열리며 아래와 같이 화면에 나타날 것이다. 여기까지 잘 되면 설치에 성공한 것이다.



### 3 텍과 TeXworks에 대하여

문서 작성 프로그램은 크게 나누어 워드 프로세서(word processor)가 아니라 조판 프로그램(typesetting program)의 두 가지 종류가 있다. 전자의 예로는 HWP, MS Word 등이 있고 후자의 예로는 Adobe InDesign과 T<sub>E</sub>X 등이 있다.

전자는 화면에서 문서를 보면서 작성하지만 후자는 작성 창(editor window)과 보기 창(viewer window)이 따로 있다. 그래서 사용하기에는 워드보다 조판 프로그램이 불편한 것이 사실이다. 그러나 조판 프로그램은 문서 형식의 일관성과 문서 내 정보처리(색인, 참고문헌 인용 등)에서 많은 장점을 가진다.

텍이라는 조판 프로그램은 여러 부분으로 이루어져 있는데 이 중에 사용자의 작업환경(혹은 인터페이스)를 제공하는 것이 TeXworks이다. 텍의 작업환경 프로그램은 TeXworks 말고도 많이 있으며, 우리가 설치한 배포판에서 사용하는 작업환경 프로그램이 바로 TeXworks이다. 이 프로그램은 무료이고 여러 좋은 기능들을 가지고 있지만 아직 개발 초기단계이므로 미흡한 점도 많다.

TeXworks는 소스 에디터와 뷰어 간에 검색/역검색이 가능하다. 즉 소스 창의 특정 위치에서 Ctrl-Click하면 뷰어 창에서 대응되는 위치가 표시된다. 역으로 뷰어 창의 특정 위치에서 Ctrl-Click하면 소스 에디터 창에서 대응되는 위치가 표시된

다. 이 기능은 앞서 만들어 보았던 `myfirst.tex`에서도 확인해 볼 수 있으나 문서의 길이가 너무 짧아 기능 확인에 썩 적당하지는 않을 것이다.

TeXworks의 단점으로는 이전에 작업했던 윈도우들의 크기와 위치를 기억하지 못한다는 것, 그리고 뷰어 창에서 페이지 이동이 다소 느리다는 것이 있다. 컴퓨터의 사양이 낮을 경우 뷰어의 페이지 이동 뿐만이 아니라 문서열기, 컴파일 등 모든 작업의 속도가 느려 불편을 느낄 수 있다. 또한 스펠링 체크 기능이 약하다는 것도 약점이다.

TeXworks에서 컴파일 작업을 하다가 에러가 발생하면 화면 좌상단의 초록색 버튼이 붉은 X표로 바뀐다. 이럴 때는 소스 파일을 제대로 고친 다음 붉은 X를 클릭하면 초록색 삼각형으로 돌아올 것이니 이때 다시 한 번 초록색 삼각형을 클릭하면 된다. 그런데 때로는 에러가 발생하여 `.tex` 소스를 고쳐도 계속 에러가 나는 수가 있다. 이것은 컴파일 도중에 생성된 보조파일들이(`.aux`, `.log` 등) 문제가 되어 그런 것이니 이들을 삭제해 준 다음에 컴파일 하면 제대로 된다. 보조 파일들을 일일이 탐색기를 열어 삭제하지 않고도 쉽게 삭제하려면 [파일 - 부수파일 삭제...]을 실행하면 된다.

TeXworks의 현재 버전은 인쇄기능이 없다. 그러므로 텍으로 작성한 PDF 문서를 인쇄하려면 어도비 뷰어, 혹은 어도비 아크로بات을 사용한다. 어도비에서 PDF 파일을 인쇄할 때 주의할 점 하나는 인쇄 윈도우의 원편에 [페이지 비율(S):]은 “없음”을 선택해야만 내가 지정한 사이즈로 인쇄되며 “인쇄 가능영역에 맞추기” 등을 선택하면 글자크기와 마진이 내가 지정한 것에서 확대/축소되어 출력된다는 사실이다.

## 4 기본 사용법

### 4.1 문서클래스와 패키지

우리가 처음으로 작성하여 컴파일 해 보았던 `myfirst.tex` 파일의 각 라인을 상세히 분석해 보자.

```
\documentclass{article}
\usepackage{kotex}
\begin{document}
나의 방정식은  $x^2+y^2=1$ 입니다.
```

```
아름다운 텍세상!
\end{document}
```

첫 라인은 문서의 클래스(class), 즉 종류를 지정한다. 텍에서 지원하는 문서의 종류는 article, report, book, letter, beamer 등으로 많은데 이 매뉴얼에서는 article과 beamer만 설명할 것이다. Article은 길이가 비교적 짧은 문서, 예를 들어 30쪽 이내의 문서 작성에 적합하다. 이 매뉴얼 자체도 article class의 문서이다. Beamer는 프리젠테이션 슬라이드를 만드는 데 사용되는 클래스이며 5절에서 더 설명할 것이다.

그 다음 라인은 한글 기능을 주는 kotex이라는 패키지(package)를 사용하기 위한 것이다. 만일 이 라인을 생략하면 영어만 있는 문서라면 문제 없이 컴파일 하여 출력파일(PDF)를 얻을 수 있으나, 한글을 사용한 문서를 컴파일 하면 출력파일에는 소스 파일의 영문 부분만 나타나고 한글 부분은 전혀 나타나지 않게 된다.

텍에서는 여러 기능을 추가하기 위한 셀 수도 없이 많은 패키지들이 있다. 이 매뉴얼에서는 이 중 몇 개를 공부할 것이다.

`\begin{document}`와 `\end{document}`는 각각 문서 내용의 시작과 끝을 나타내는 명령(command)이다. 명령은 특수한 기능을 가지는 문자열로서 자기 자신은 출력에 그대로 나타나지는 않으나 문서의 출력에 어떤 방법으로 관여하는 것이다. 명령에 대해서는 조금 후에 다시 설명하겠다.

## 4.2 코멘트, 수식, 특수문자, 빈칸, 줄바꿈, 명령

다음과 같은 파일을 작성하여 my2nd.tex으로 저장하고 컴파일 한다.

```
\documentclass{article}
%\usepackage{kotex}
\begin{document}
나의 방정식은  $x^2+y^2=1$ 입니다.

Oh      beautiful \TeX\ World 아름다운 텍세상! %Oh yeah!
\end{document}
```

이 문서는 kotex 패키지를 사용하지 않으므로 출력 PDF 파일에 한글은 나타나지 않고 영어만 보일 것이다. 두 번째 라인의 맨 첫 글자가 %임에 주목하라. TeXworks 에디터에서는 이 라인이 붉은 색으로 나타날 것이다. 이 라인은 소스 파일에 존재하기는 하나 그냥 무시된다. 마지막 라인에의 %Oh yeah! 부분도 에디터에서는 붉게 나타나고 출력물에는 나타나지 않는다.

어떤 라인에 % 기호가 나타난다면 그 기호부터 시작하여 무시된다. 이 무시되는 부분을 코멘트(comment)라고 부른다.

TeXworks 에디터에 코멘트 관련 편리한 기능이 있다. 여러 개의 라인을 한꺼번에 코멘트 처리할 때 원하는 라인을 선택한 후 Shift-Ctrl-] 하면 모든 라인이 %

로 코멘트 처리 된다. 이것의 역은 Shift-Ctrl-[이다.

수식은 두 개의 \$ 기호 사이에 넣으면 된다. 수식에서 ^는 위첨자(superscript)를 나타내기 위한 것이다. 아래첨자(subscript)는 \_를 사용하여 나타낼 수 있다.

수식을 나타낼 때 사용하는 기호 \$를 특수문자(*special character*)라고 한다. 특수문자는 특별한 기능을 가지는 문자로서 \와 %도 특수문자이다. 이러한 특수문자를 출력물에 나타내려면 통상 그 문자 앞에 \를 쓰면 된다.

위 문서의 마지막 부분의 Oh beautiful에 빈칸이 여러 개 들어 있는데 출력물에서는 빈 칸 1개로 처리 되었음을 볼 수 있을 것이다. 그리고 “나의 방정식은” 라인과 “Oh beautiful” 라인 사이에 빈 줄이 하나 들어 있는데 이것 역시 무시되었음을 볼 수 있을 것이다. 텍을 사용할 때는 아래의 사실들을 명심해야 한다.

- (1) 빈칸 여러 개는 하나와 같다.
- (2) 줄바꿈 하나는 빈칸 하나와 같다.
- (3) 줄바꿈 2개 이상은 모두 줄바꿈 2개와 같으며 출력물에서는 문단바꿈으로 나타난다.

위에서 ‘문단바꿈’이라는 용어를 썼는데, 문단바꿈(new paragraph)은 줄바꿈(new line)과 비슷하나 들여쓰기(indent)가 있다는 점이 다르다. 실은 문단간격을 줄 수 있다는 점에서도 다른데 이에 대해서는 다른 곳에서 설명하겠다.

소스에 있는 \TeX은 출력물에서는 좀 더 멋을 부려 T<sub>E</sub>X으로 나타난다. 소스가 출력물에 그대로 나타나지 않았으니 이것도 명령(command)의 일종이다. 모든 명령은 backslash로 시작하며 영문 대문자/소문자로 이루어진 문자열이 따라 붙는다. 숫자와 특수문자는 명령 문자열에 사용할 수 없다. 단, backslash와 한 개의 특수문자로 이루어진 명령이 몇 개 존재한다. 예를 들어 \%는 % 기호를 출력하는 명령이다.

그런데 \TeX\와 같이 명령 \TeX 뒤에 \를 넣은 것은 무엇인가? 이 backslash를 제거하면 빈칸이 무시되고 ‘T<sub>E</sub>XWorld’와 같이 나타난다. (컴파일 해서 확인해 볼 것.) 이것은 명령 뒤의 빈칸(들)은 무시되기 때문이며 이것을 방지하기 위하여 \을 넣어 준 것이다.

어떤 명령은 인수(argument)를 사용한다. 예를 들어 \emph{강조, emphasize}는 강조, emphasize로 나타나는데 여기서 명령 \emph의 인수가 “강조, emphasize”인 것이다.

\begin과 \end는 환경(environment)의 시작과 끝을 나타내는 명령이며 대단히 빈번히 사용된다. 아래의 예를 보라.



```

\documentclass{article}
\usepackage{kotex}
\begin{document}
항목들을 나열할 때 쓰는 환경을 사용해 보았다.
\begin{itemize}
\item 첫 번째 항목
\item 두 번째 항목
\end{itemize}
그리고 또,
\begin{center}
이건 가운데 위치하게 될 거임.
\end{center}
\end{document}

```

텍에서 사용하는 명령은 무수히 많으며 이 중 중요한 것들은 암기해야 한다. 사용자가 자신이 원하는 명령을 만들어 사용할 수도 있으며 이에 대한 설명은 다음으로 미룬다.

이번에는 아래의 소스를 컴파일 해 보자.

```

\documentclass{article}
\usepackage{kotex}
\begin{document}
수식을 쓰려면 수식의 전후에 \$를 쓰면 된다.
그런데 backslash 기호 \ 는 어떻게 나타내야 하는 건가?
\end{document}

```

특수문자 \$ 앞에 \를 사용했으므로 \$가 나타난 것은 예상대로이다. 그런데 \를 나타내기 위하여 \를 연이어 2개를 사용했는데 나타난 것은 \가 아니라 줄바꿈이다. \\는 다양한 상황에서 줄바꿈에 이용되는 중요한 명령이니 꼭 기억해 두어야 한다.

마지막으로 아래의 소스를 컴파일 해 보라. 그리고 수식 작성 명령들의 기능을 추리해 보라.

```

\documentclass{article}
\usepackage{kotex}
\linespread{1.25}\selectfont
\begin{document}
수식을 쓰려면 수식 부분의 전후에 \$를 쓰면 된다. 그리고 수식을 크게,
단독으로 나타내려면 수식의 전후에 \$\$를 사용한다. 이렇게 말이다.
\[\sum_{i=1}^n \frac{1}{2^n} = 1\] 분수를 나타내는 법, 시그마(합
기호)를 나타내는 법을 알겠는가? 이제 아래의 수식만 이해하면 가장
기본적인 명령은 익힌 셈이다. 중괄호는 묶기(grouping)에 쓰이는 중요한

```

특수 기호이다. 중괄호 자체를 수식에서 나타내려면 아래와 같이 한다.

집합  $\{n, n+1, \ldots, m\}$ 의 원소의 개수는  $m+n-1$ 이다.

이제 아래의 수식만 쓸 수 있다면 기본적인 수식 관련 명령은 다 익힌 셈이다.  
$$\frac{x_1+x_2+\cdots+x_n}{n} \geq \sqrt{x_1 \cdot x_2 \cdots x_n}$$

이 소스에는 수식 관련 명령들 외에도 꼭 배워야 할 것이 하나 있다. 그것은 줄간격을 1.25배 늘리는 역할을 하는 `\linespread{1.25}\selectfont` 명령이다. 원래 텍 시스템은 영어를 염두에 두고 만든 것이다. 그런데 한글에서는 줄간격이 영문에 서보다 1.2배에서 1.4배 정도 더 넓어야 보기가 좋다. 그래서 위 문서에서 이 명령을 사용한 것이다.

### 4.3 타이틀 페이지, 목차

문서에서 타이틀 페이지는 문서의 제목, 저자의 이름, 작성일자 등의 정보를 보여주는 부분을 뜻한다. 사실 `article` 클래스의 문서에서는 타이틀이 별도의 독립된 페이지를 가지지 않으므로 타이틀 페이지라고 하는 표현은 정확하지 않고 그냥 ‘타이틀’이라고 불러야 할지도 모르겠다.

타이틀 페이지를 만들려면 소스 문서의 첫 부분을 아래와 같이 하면 된다.

```
\documentclass{article}
\usepackage{kotex}
\linespread{1.25}\selectfont
\title{ko.\TeX의 기본}
\author{Joohee Jeong}
\date{\today}
\begin{document}
\maketitle
...
```

제목은 `\title` 명령을 사용하고, 저자는 `\author` 명령을 사용한다. 그리고 문서작성일자는 ‘2011년 4월’과 같이 넣어도 되고, `\today` 명령을 사용하여 오늘 날짜를 넣어도 된다. 그리고 `\begin{document}` 바로 아래에 `\maketitle` 명령을 넣는다.

텍 소스 파일에서 `\begin{document}` 이전 부분을 전문(*preamble*)이라고 한다. 전문에서는 문서의 클래스 지정, 사용 패키지 지정, 타이틀 페이지 정보 입력, 사용자 정의 명령(이건 아직 공부하지 않았음), 기타 문서 전반 모양새를 지정하는 명령들이 들어간다.

텍은 훌륭한 목차 자동작성 기능을 가지고 있다. 이 매뉴얼에 사용된 목차 관련 명령들만 모아 보면 다음과 같다.

```
\section{\TeX{}을 공부하려면}
\section{설치, 그리고 첫 사용}
\section{텍과 TeXworks에 대하여}
\section{기본 사용법}
\subsection{문서 클래스와 패키지}
\subsection{코멘트, 수식, 특수문자, 빈칸, 줄바꿈, 명령}
...
```

위에 보인 목차 관련 명령들 사이에 본문 내용을 넣으면 된다. 단, 출력물에 목차를 나타내기 위하여는 컴파일을 2번 해야 한다는 점을 알아야 할 것이다. (한 번 컴파일 하면 .toc 파일이 생성된다. 한 번 더 컴파일하면 이 파일에 들어있는 정보로부터 목차를 만들어 낸다.)

`\TeX` 명령 뒤에 `{}`는 사실상 생략해도 된다. 원래 `{}`는 텍의 명령을 그 뒤에 나오는 문자열과 분리하는 기능을 가지는데, 커맨드 뒤에 바로 한글 문자열이 나오는 것이 별 문제를 일으키지는 않지만 그래도 커맨드와 일반 문자열을 확실하게 분리한다는 의미에서 사용해 보았다. 같은 목적으로 `{\TeX}`을 공부하려면을 사용해도 된다.

## 4.4 그래픽 이미지

그래픽 파일 삽입은 png, jpg, pdf를 지원한다. `install_file_logo.png` 파일을 마련하여 아래에 보인 소스 파일과 같은 디렉토리에 넣고 컴파일 해 보라.

```
\documentclass{article}
\usepackage{kotex,graphicx}
\linespread{1.25}\selectfont
\begin{document}
텍 문서에 그래픽 파일을 삽입하려면 \verb+\includegraphics+ 명령을
사용하면 된다. \includegraphics{install_file_logo} 확장자는
생략해야 한다. 그리고 \texttt{graphicx} 패키지를 사용해야 한다.
```

그래픽 이미지가 인라인(inline)으로 들어가지 않고, 별도의 독립된 문단을 차지하게 할 수도 있다.

```
\begin{center}
\includegraphics[scale=1.5]{install_file_logo}
\end{center}
```

두번째 그림은 scale 옵션을 주어 크기를 1.5배 확대한 것이다.  
`\end{document}`

위의 예에서는 그래픽 이미지 관련 명령 말고도 `\verb`와 `\texttt` 명령이 나온다. 전자와 후자는 둘 다 타자폰트를 사용한다는 점에서 같다. 전자는 모든 특수문자와 커맨드를 무시하고 그대로 보여준다는 점에서 후자와 다르다.

위의 예에서 사용한 그래픽 파일은 이것을 클릭하여 다운 받을 수 있다.

이것으로 article class 문서의 작성의 가장 기본적인 사항들을 알아 보았다. 이제 2쪽에서 소개한 “ $\text{\LaTeX} 2_{\epsilon}$  입문”의 1장~3장을 읽기 바란다. 텍에서 꼭 필요하면 서도 이해하기 까다로운 부분은 이 매뉴얼에서 거의 모두 설명하였으므로 “ $\text{\LaTeX} 2_{\epsilon}$  입문”은 이제 거의 소설책 읽듯이 술술 읽을 수 있을 것이다.

## 5 프리젠테이션 슬라이드

프리젠테이션 슬라이드 제작을 위한 소프트웨어 중 가장 널리 쓰이는 것은 아마도 MS PowerPoint 일 것이다. 그런데 수식을 많이 사용하는 프리젠테이션의 경우 텍의 Beamer 클래스를 사용하는 것이 더 좋다고 생각한다. 비머의 기능과 장점은 사용법을 공부하면서 차차 알게 될 것이다.

비머 파일의 예를 하나 자세히 들여다 보면서 공부하기로 하자. 모든 파일은 [ko. \$\text{\TeX}\$ 과 Beamer.zip](#)의 압축을 풀어 얻을 수 있다.

### 5.1 포함 파일과 테마

샘플 비머 파일 `my_beamer_sample.tex`의 전문을 보면서 공부하자. 전반부와 후반부의 2 부분으로 나누어 보기로 한다.

```
\documentclass[hyperref={pdfTeX,unicode},xcolor={svgnames},t,12pt]{beamer}
\input{my_beamer_hangul}

\usetheme{Darmstadt}
    % Choose from Darmstadt/Montpellier/Warsaw
\usecolortheme[named=SteelBlue]{structure}
    % Choose [named] from Olive/SteelBlue/LightSteelBlue/
    %   Goldenrod/Tan/Gray etc.
%\setbeamertheme{navigation symbols}{}
    % comment out to hide navigation symbols at bottom
%\useoutertheme{infolines}
    % shows (short form of) author,title,date and current page
```

```

% in bottom strip.
% top strip changes to show section:subsection
\newcommand{\overlayopacity}{15} % percent
\newcommand{\myemphcolor}{Red} % color for \hemph, \bemph

\input{my_beamer_style}
\input{my_beamer_math}

```

제일 첫 줄은 문서 클래스를 beamer로 지정하면서 여러 옵션을 넣은 것이다. 둘째 줄은 내가 미리 준비해 놓은 파일 my\_beamer\_hangul.tex을 포함시키는 것이다. 모든 비머 소스 파일의 전문에 항상 넣어야 할 내용은 별도의 파일로 마련해 놓고 이렇게 input 하는 것이 소스가 깔끔해서 좋다. 위에 보인 소스의 맨 아래에서 두 개의 파일 my\_beamer\_style.tex과 my\_beamer\_math.tex을 추가로 input 하고 있다.

my\_beamer\_hangul.tex은 한글 비머 파일에 필요한 명령들을 포함한 짧은 파일이다. (빈 줄을 제외하면 8줄로 이루어져 있다.) my\_beamer\_style.tex은 비머 문서의 모양새를 위하여 내가 만든 명령들을 이루어져 있다. 대략 30줄 정도이다. my\_beamer\_math.tex은 비머 문서에서 수식 작성에 내가 흔히 사용하는 명령들을 모은 것이다.

이상 3개의 포함 파일 중에서 my\_beamer\_hangul.tex를 제외하고는 나 개인의 취향이 많이 들어 있으므로 독자들은 텍의 지식을 충분히 갖춘 후에는 두 파일 my\_beamer\_style.tex과 my\_beamer\_math.tex를 자신이 원하는 대로 알맞게 고쳐 사용해야 할 것이다.

포함파일은 소스파일과 같은 디렉토리에 두면 컴파일 된다. 그러나 여러 개의 비머 문서를 만들어 관리할 때 각 디렉토리마다 포함파일 3개를 꼬박꼬박 넣어 주는 것은 별로 현명한 일이 아니다. TeXworks에는 이렇게 자주 사용하는 포함파일들을 위한 디렉토리가 있으므로 거기에 넣어 두면 다음부터는 소스파일 디렉토리에 포함파일을 넣을 필요가 없어서 편리하다. 이 디렉토리는

```
C:/usr/texlive/texmf-local/tex/latex/local
```

이다. 포함파일들을 이리로 옮겨 넣은 후에는 한 가지 작업을 해야 하는데 그것은 TeXhash이다. 이를 위하여는 작업표시줄에서 [시작 - 모든 프로그램 - ko.TeX Live 2010 - TeX Live Manager]를 실행시키고 프로그램 화면에서 [Actions - Update filename database]해 주면 된다. 이 프로그램은 화면을 띄우고, 실행하고, 화면을 닫는 데에 시간이 좀 걸리니 느긋하게 기다려야 할 것이다.

위 소스의 3번째 줄 `\usetheme{Darmstadt}`는 출력물의 모양새 테마로 Darmstadt를 지정하고 있다. 바로 아래의 코멘트를 보면 Darmstadt 외의 테마로 Montpellier와 Warsaw를 선택해도 된다고 말해 주고 있다. 이 외에도 사용자가 선택할 수 있는 테마가 많이 있으며 상세한 것은 이 문서의 첫 부분에서 소개한 the BEAMER class User Guide를 참조하길 바란다.

그 아래의 `\usecolortheme[named=...]`는 비머 문서의 색깔 테마를 지정하고 있다. 텍이 인식하는 임의의 색깔을 named의 값으로 지정할 수 있으며, 나의 추천 색깔인 Olive, SteelBlue 등을 코멘트에 보였다.

그 아래의 `\setbeamertheme{...}`와 `\useoutertheme{infolines}`의 기능은 코멘트에서 간단하게 설명하고 있는데, 확실히 알기 위하여는 현재 코멘트 아웃되어 있는 것을 해제하고 컴파일 했을 때 출력물에 어떤 변화가 일어나는지를 관찰하는 것이 가장 좋다.

그 아래의 두 줄은 독자가 텍의 지식을 더 갖춘 후에 문자열 `overlayopacity`와 `myemphcolor`를 `my_beamer_style.tex`에서 찾아 보면 의미를 알 수 있을 것이다. 여기서는 일단 그냥 넘어 가기로 한다.

## 5.2 타이틀 페이지

전문의 나머지 부분을 아래에 보였다.

```
%% my personal commands

\newcommand{\kotex}{ko.\TeX}
\newcommand{\latexe}{\LaTeX\hspace{0.3ex}2\hspace{0.1ex}%
  \raisebox{-0.3ex}{${\varepsilon}$}}

%% titlepage
\title{\kotex{} Live and Beamer}
\subtitle{한글 \TeX\ 사용 자 환경}
\author{\textsf{정 주 희 (Jeong, Joohee)}}

\institute[\textsf{K.N.U.}]{\textsf{Kyungpook National University}}
\date[\textsf{\today}]{\textsf{\today. 자연과학관 105-2}}

%% end of preamble

\begin{document}
```

```

\begin{frame}
  \titlepage
\end{frame}

\begin{frame}
\frametitle{목 차}
\tableofcontents
\end{frame}

\section{\TeX{}을 공부 하려면}

\begin{frame}
\frametitle{참고 문헌}
텍의 \uline{ 기초 } 지식을 위한 참고 문헌
\begin{enumerate}[(1)]\linespread{1.1}\selectfont
\item \hrefc{http://proofmood.com/~jhjeong/mybox/ ...}
\item ...
\item ...
\item ...
\end{enumerate}
\end{frame}

```

위의 소스의 맨 처음에는 사용자 정의 명령 2 개가 나와 있다. `\newcommand{..}{..}`의 구문을 사용하는데 그냥 척 봐도 대략 어떻게 되는 건지 이해가 될 것이다. 상세한 것은 이 문서의 첫 부분에 소개한 참고문헌들을 참조하길 바란다.

그 다음에 나오는 `\title`과 `\subtitle`는 각각 문서의 제목과 부제이다. `\author`는 저자의 이름이다. 그런데 여기서 `\textsf`라는 명령이 쓰였는데 이것은 고딕체를 사용하라는 의미이다.

`\institute`는 저자의 소속 기관을 지정하는 명령이다. 여기서 알아야 할 것은 소속 기관을 `\institute[짧은 버전]{긴 버전}` 형태로 지정할 수 있다는 것이다. 바로 아래의 `\date`도 짧은 버전과 긴 버전을 지정하였음을 볼 수 있다.

짧은 버전은 앞서 나왔던 `\useoutertheme{infoclines}`의 코멘트를 해제하고 컴파일 하여 보면 화면의 하단에 나타나니 실제 컴파일을 실행, 관찰하여 확실하게 알아 두길 바란다.

### 5.3 목차, 프레임과 슬라이드, 오버레이

프리젠테이션 슬라이드에는 오버레이라는 개념이 있다. 과거에 많이 사용하던 Overhead Projector(OHP)에서는 하나의 슬라이드 필름을 화면을 보여 주는 상태에서 다른 필름을 겹쳐 놓아서(overlay 하여) 보여주곤 했는데 이것이 프리젠테이션에서는 상당히 유용하다. 비머 클래스는 이런 효과를 쉽게 얻을 수 있는 방법을 제공한다.

비머에서 예를 들어 어떤 페이지에서 오버레이를 하나 얹어 사용하게 되면 이 페이지에서는 한 개의 프레임(*frame*)이 두 개의 슬라이드(*slide*)로 구성되어 있다고 말한다. 만일 오버레이를 두 개 얹었다면 이때는 한 개의 프레임이 3개의 슬라이드로 이루어지게 되는 것이다. 그러니까 비머에서는 ‘페이지’라는 용어는 혼동을 피하기 위하여 사용하지 않는다.

비머의 첫 프레임은 타이틀페이지이다. 둘째 프레임은 목차이다. 이 프레임에서는 `\frametitle` 명령을 이용하여 프레임의 제목을 주어야 한다. 통상 ‘목차’, 혹은 ‘차례’, 혹은 영어로 ‘Outline’ 등의 제목을 준다.

비머 문서는 절(section)과 부절(subsection)로 이루어지며 이것들이 목차를 구성함은 article 클래스 문서의 경우와 같다. 비머가 article과 다른 점은 각 프레임마다 또 `frametitle`을 주어야 한다는 것이다. 처음에 비머를 사용할 때는 아직 익숙하지 않아서 프레임 제목 주는 것이 좀 귀찮을 수 있다.

`my_beamer_sample.pdf`의 3쪽에 (타이틀 페이지와 목차를 제외한) 첫 프레임이 있다. 소스에서 먼저 눈에 띄는 것은 `\uline` 명령인데 이것은 밑줄을 긋는 기능을 가진다. 그 다음이 중요하므로 여기에 소스를 일부 옮겨 보았다.

```
\begin{enumerate}[(1)]\linespread{1.1}\selectfont
\item \hrefc{http://proofmood.com/...}{...} ...
\item \hrefc{http://proofmood.com/...}{\alert<2>\latexe{} 입문}} by ...
\item ...
\item ...
\end{enumerate}
```

프리젠테이션 문서는 주로 개조식(個條式)<sup>†</sup>을 사용하는데 `ko.TEX`은 개조식 문서를 만드는 좋은 도구들을 제공한다. `enumerate` 환경은 다양한 방법으로 각 항의 시작 부분의 모양을 설정할 수 있다. 이 문서에서는 (1), (2), ... 와 같이 번호를 붙이므로 [(1)]를 쓴 것이다. [(1)] 대신에 [①], [②], [③], [㉿], [\itshape (i)] 등을 넣고 출력물이 어떻게 바뀌는지 실험해 보길 바란다. 마지막의 예에서 사용한 `\itshape`는 *italic shape*를 의미한다. 항목의 머리에 번호를 붙이지 않으려면

<sup>†</sup>글을 쓸 때에, 앞에 번호를 붙여 가며 짧게 끊어서 중요한 요점이나 단어를 나열하는 방식.



[ $\bullet$ ] 등을 사용하면 된다.

그 다음은 링크에 대해서 알아 보자. 텍은(pdf $\text{\LaTeX}$ )은 문서 내 링크는 물론 인터넷 연결 링크까지 제공한다. 여기서는 후자만 설명할 것인데 다음과 같은 구문을 사용하면 간단히 구현된다.

```
\href{http://proofmood.com/..}{링크에보여질문자열}
```

위의 예에서는 원래 인터넷 링크를 만들 때 사용하는 `\href` 명령 대신 `\hrefc`를 사용했는데 후자는 내가 만들어 `my_beamer_style.tex`에 넣어 둔 명령이다. 이것은 `\href`와 같으나 다만 링크 문자열의 색깔이 디폴트인 `black` 대신 `\hrefc steelblue`로 나타나는 것이 다르다.

두 번째 항목의 `\alert<2>\{..\}`에서 `<2>`만 삭제하고 컴파일 하면 ... 부분이 붉은 색으로 나타난다. 이렇게 뒤따르는 중괄호 내부의 문자열을 붉은 색으로 만드는 것이 `\alert` 명령의 기능이다. 그런데 여기에 `<2>`를 넣음으로써 오버레이가 만들어진다. 즉 이 프레임에는 2개의 슬라이드로 이루어지며, `\alert` 명령의 효과는 둘째 슬라이드에만 나타나게 되는 것이다.

오버레이를 더 정확히 이해하기 위하여 소스의 전문에서 `\useoutertheme{..}` 앞의 `%`를 삭제하여 이 명령을 활성화 시킨 다음 컴파일 해 보라. 화면의 맨 아래에 가는 띠가 나타나며 타이틀 페이지의 정보(짧은 버전)와 아울러 맨 오른쪽 구석에는 쪽번호(프레임 번호)  $n$ 이 문서 전체의 프레임의 개수  $N$ 과 함께  $n/N$  형태로 나타날 것이다.

다음 번 프레임은 제목에 `\ftcont`라는 명령을 사용했는데 이건 그냥 ‘continued’라는 문자열을 만들어 주는 간단한 명령으로서 역시 `my_beamer_style.tex`에 정의되어 있다. 비머 문서를 작성하다 보면 프레임의 제목을 앞의 프레임과 똑 같이 쓰고 싶을 때가 많은데, 두 프레임에 완전히 동일한 제목을 사용하는 것 보다는 이렇게 동일한 제목을 사용하되 그 뒤에 ‘continued’를 덧붙이는 것이 괜찮아 보인다.

5번 프레임에 오버레이를 사용하는 또 하나의 예를 보였다. 여기서 사용한 명령은 `\uncover<2>\{..\}`와 `\visible<2>\{..\}`의 2개인데 전자는 첫 슬라이드에서는 희미하게 보여주다가 2번째 슬라이드에서 제대로 보여주는 것이고, 후자는 첫 슬라이드에서는 아예 전혀 보여주지 않다가 2번째 슬라이드에서만 보여주는 것이다. 희미하게 보여줄 때의 불투명도는 전문에서 현재 `\overlyopacity`에서 15로(15%로) 지정되어 있다.

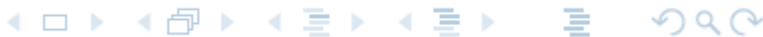
6번 프레임에 다시 새로운 오버레이 방법을 사용했다. 예를 들어 `<3>`는 3번째 슬라이드에서만 보여주라는 의미인 반면에 `<3->`는 3번째 슬라이드에서 시작하여 그 이후의 모든 슬라이드에서 보여주라는 뜻이다.

8번 프레임에서 사용한 `\bemph{한글 용어}{영어 용어}` 명령의 기능은 출력물에서 보면 알 수 있다. 한글과 영어 문자열을 각각 적당한 폰트로 바꾸어 주고 색깔은 강조색깔로 나타내는 것이다. 또한 영어 문자열 전후에 괄호를 둘러 준다. 현재의 강조색깔은 전문의 `\myemphcolor`에서 Red로 지정되어 있다.

9번, 10번, 11번 프레임에는 `\bigskip`이라는 명령이 사용되었다. 이 명령은 문단 사이의 간격을 띄우는 기능을 가진다. 비슷한 것으로 `\medskip`, `\smallskip` 등의 명령도 있으니 사용해 보기 바란다.

11번 프레임에 또 하나의 오버레이 요소가 사용되었다. 여기서 사용한 `\pause`는 `\uncover`와 같이 뒤 따르는 내용을 첫 슬라이드에서는 희미하게 보여 주다가 다음 슬라이드에서 정상으로 보여준다. 그러나 `\uncover`와는 달리 `<2->`와 같은 것이 필요 없으며, 뒤 따르는 내용을 중괄호 `{ ... }`로 두르지 않아도 된다.

비머 출력 화면의 하단에는 아래와 같은 네비게이션 버튼들이 보인다. 이 버튼들이 나타나지 않게 하려면 전문에서 `setbeamertemplate{navi..}{}`의 코멘트 기호 %를 제거하면 된다.



이 버튼들은 TeXworks의 PDF 뷰어에서는 때로는 작동이 잘 안되기도 하지만 어도비 뷰어에서는 확실하게 작동한다.

맨 왼쪽의 화살표(사실은 삼각형이지만 화살표라고 부르겠다.)를 클릭하면 이전 슬라이드로 이동한다. 그 다음의 직사각형을 클릭하면 팝업창이 뜨며 여기에 이동하고 싶은 대상(target) 프레임 번호를 넣고 [확인]을 클릭하면 그 프레임의 첫번째 슬라이드로 이동한다. 그 다음의 화살표는 다음 슬라이드로 이동하는 것이다.

그 다음 3개의 버튼 중 왼쪽 화살표는 이전 프레임(의 맨 뒤 슬라이드)으로 이동, 오른쪽 화살표는 다음 프레임(의 맨 처음 슬라이드)으로 이동하는 것이다. 직사각형을 겹쳐 놓은 것 같이 보이는 버튼은 현재 프레임의 맨 뒤 슬라이드로 이동한다.

그 다음 3개의 버튼은 부절(subsection)에 대한 것이고, 또 그 다음 3개의 버튼은 절(section)에 대한 것이니 정확한 의미는 프레임의 경우와 비교해서 생각하면 곧 알 수 있을 것이다.

그 다음에는 화살표는 없고 진한 색깔의 책을 여러 권 겹쳐 놓은 듯한 버튼이 있는데, 이 버튼, 혹은 그 오른쪽 빈 곳을 클릭하면 문서의 맨 뒤로 이동하고, 이 버튼의 왼편 빈 곳을 클릭하면 문서의 맨 처음으로 간다.

그 다음에 왼편으로 회전하는 화살표 버튼은 직전에 본 슬라이드로 이동한다. 오른쪽으로 회전하는 화살표 버튼은 왼편 회전 화살표 버튼의 역이다. 그러니까 웹브

라우저의 뒤로 (backward)/ 앞으로 (forward) 버튼과 같다고 보면 된다.

마지막으로 이 두 회전하는 화살표 사이의 돋보기 같이 생긴 버튼은 문자열 검색 버튼이다. 영문은 물론 한글 문자열의 검색도 잘 된다.

————— ○ ○ —————

이 매뉴얼에는 나와 있지 않지만 꼭 알아 두어야 할 비머 관련 지식으로 `tabluar`, `columns` 및 `block(theorem, lemma 등)` 환경이 있다.

`my_beamer_style.prg`의 12번 프레임 이후에 이들 환경의 사용 예가 있으니 `my_beamer_style.tex`의 소스와 비교하면 어떤 개념인지 알 수 있을 것이다. 이에 대한 간단한 설명을 비머에 대한 알기 쉬운 매뉴얼인 [A Beamer Tutorial in Beamer](#)에서 찾아 볼 수 있다.

————— 끝 —————