

# Essential L<sup>A</sup>T<sub>E</sub>X++ 개정한글판

January 1994 — 2005, 2015 (한글판)

Jon Warbrick (원저)

David Carlisle, Michel Goossens,

Sebastian Raetz, Adrian Clark (추가)

김재우 · 김강수 (옮김)    김강수 (개정)



## 1 개요

본 문서는 L<sup>A</sup>T<sub>E</sub>X<sup>1</sup> 문서준비시스템을 이용하기 위해서 기본적으로 알아야 할 내용에 대해 다룬다. 기초적인 내용만 언급할 것이고 상세한 내용은 대부분 생략하였다. 사실 이 정도 분량의 문서로 필요한 모든 사항을 포함할 수는 없으며, 프로그램을 본격적으로 사용하기 위해서는 보다 완전한 참고서를 참고하여야 한다. 이 글만 읽고 복잡한 문서를 모두 만족스럽게 작성할 수 없을지 모른다. 그러나 기본적인 요소를 갖춘 L<sup>A</sup>T<sub>E</sub>X 문서를 작성하고 L<sup>A</sup>T<sub>E</sub>X의 동작을 이해하는 데는 부족함이 없을 것이다.

2015년 현재 1990년대에 저술된 Leslie Lamport의 *The L<sup>A</sup>T<sub>E</sub>X User's Guide and Reference Manual* [1]만을 가지고 L<sup>A</sup>T<sub>E</sub>X의 모든 기능을 전부 이해하기는 어렵다. Lamport가 만든 L<sup>A</sup>T<sub>E</sub>X에서 현재까지 수십 년 동안 눈부신 발전이 있었다. 그러나 이 책은 L<sup>A</sup>T<sub>E</sub>X의 설계와 기본 개념을 이해하기 위해서는 한번쯤 읽어보아야 한다. 일반사용자로서는 온라인 상에서 구할 수 있는 매뉴얼로도 충분히 공부를 시작할 수 있다. 예를 들면 위키문헌의 L<sup>A</sup>T<sub>E</sub>X은 매우 좋은 출발점이다. 이밖에 T<sub>E</sub>X Live와 함께 설치되는 다음 문서들을 참고하기 바란다.

- `lshort`. “The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X,” by Tobias Oetiker et. al.
- `novices`. “L<sup>A</sup>T<sub>E</sub>X for Complete Novices,” by Nicola Talbot.
- `latex4wp`. “L<sup>A</sup>T<sub>E</sub>X for Word Processor Users,” by Guido Gonzato.

이 글은 L<sup>A</sup>T<sub>E</sub>X 문서 작성에 대한 것이므로 T<sub>E</sub>X과 L<sup>A</sup>T<sub>E</sub>X 시스템의 설치에 관한 사항은 생략한다. 즉, 이 문서를 읽는 사용자가 이미 T<sub>E</sub>X Live를 설치하고 적절한 에디터로 문서를 작성할 준비가 되었다고 가정한다.

## 2 L<sup>A</sup>T<sub>E</sub>X은 어떻게 동작하는가?

L<sup>A</sup>T<sub>E</sub>X을 사용하기 위해서는, 사용자는 조판할 본문과 본문을 사용자가 원하는 모양이 어떤 것인지를 L<sup>A</sup>T<sub>E</sub>X에게 알려주는 지시문 (mark-up)을 모두 담고 있는 파일을 작성하여야 한다. 이 파일은 보통 시스템 에디터를 이용하여 작성한다. 파일의 이름은 필요한 대로 정하되 파일 유형을 식별할

<sup>1</sup>보다 정확히 말해서 1994년 1월 또는 그 이후에 발표된 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>를 말한다.

수 있도록 “.tex” 확장자를 붙인다. 이제 이 파일에 대하여 `pdflatex` 또는 `xelatex` 명령을 다음과 같이 실행하면

```
$ pdflatex sample.tex
```

같은 파일이름에 확장자가 “.tex”에서 “.pdf”로 바뀐 새로운 파일이 생겨난다.

L<sup>A</sup>T<sub>E</sub>X에서 사용자는 문서가 어떻게 배치되어야 할지를 일일이 지시하는 것이 아니라 문서의 논리적 구조를 기술하면 된다. 예를 들자면, 본문 중에서 강조되는 인용문을 이러한 논리적 구조의 한 요소로 생각할 수 있다. 이러한 인용문은 기본 본문과 구별될 수 있도록 적절한 모양으로 조판하는 것이 일반적이다. 옛날의 인쇄 출판업에서 조판공은 원고에서 인용문으로 표시된 부분을 식별하고 그에 따라 문단을 짜는 등의 일을 하였지만, L<sup>A</sup>T<sub>E</sub>X은 단순히 프로그램에 불과하므로 원고작성자가 이를 지시하여야 한다. 따라서 사용자가 인용문임을 선언하고 L<sup>A</sup>T<sub>E</sub>X이 그에 따라 인용문을 정확히 조판하도록 해주는 L<sup>A</sup>T<sub>E</sub>X 명령어가 존재하는 것이다.

L<sup>A</sup>T<sub>E</sub>X의 출발점은 *documentclass*이다. 이것은 문서가 정확히 어떤 모양으로 조판될 것인지, 인용문과 같은 논리구조를 어떻게 조판할 것인지에 대한 설정을 담고 있다. 표준 *documentclass*인 *article*, *book*, *report* 등이 제공된다. 한편 사용자는 수식 등 문서 내의 특정 논리적 구조의 조판 형식 (*format*)을 정의함으로써 클래스의 표준 설정을 보충할 수 있다. 완전히 새로운 스타일을 작성하려면 조판에 대한 기본적 원칙을 잘 알고 있어야 하는 것이 당연하지만 표준 *documentclass*를 수정하거나 새로 작성하는 것 또한 가능하다. 다시 말하면, 문서작성자는 단지 논리적 구조만을 서술한다. 그리고 그 논리적 구조의 형태적 구현은 클래스와 스타일이 담당하는 것이다. 일반적으로 말해서 표준 클래스의 포맷이 마음에 들지 않는다면 선호하는 포맷을 구현해주는 클래스를 차용하거나 자신(또는 누군가)이 논리 구조의 구현 형태를 제공하면 된다.

문서의 외형이 아니라 논리적 구조에 집중해야 할 이유는 많이 있다. 논리적 구조가 아니라 외형을 그때그때 임의적으로 설정하여 문서의 심미적인 질을 높일 수 있다는 잘못된 믿음—문서 디자인의 첫째 기능은 아름다움이 아니라 읽기 쉬운 문서를 만드는 데 있다는 점을 알아야 한다—에서 오는 기본적인 조판 실수를 방지할 수 있다. 또한 문서 내의, 예컨대 모든 인용문의 모양을 변경하고자 할 때에도 인용문 스타일의 정의만을 변경하면 되므로 보다 유연하다. 가장 중요한 점은 논리적 디자인은 보다 좋은 글을 쓸 수 있도록 해준다는 점이다. 비주얼 시스템은 구조에 집중하는 것에 비해 시각적 효과를 손쉽게 변경할 수 있지만 논리적 디자인은 글쓰기에 좀더 집중할 수 있도록 해주어, 형태에 관심을 갖느라고 내용을 희생하는 일이 없도록 한다.

### 3 샘플 L<sup>A</sup>T<sub>E</sub>X 파일

그림 1을 살펴보자. 이것은 표준 L<sup>A</sup>T<sub>E</sub>X 예제 파일 `small2e.tex`이다. L<sup>A</sup>T<sub>E</sub>X에 기본적으로 들어 있는 파일이므로 자신의 사본 하나를 마련하여 검토하는 것이 좋다. 파일의 위치는 `kpsewhich small2e.tex` 명령으로 확인할 수 있다.

좌측에 있는 행번호는 파일의 일부는 아니지만, 참조를 손쉽게 하기 위하여 첨가하였다. 이 파일의 처리 결과가 그림 2에 있다.

이 샘플에서 주요한 점을 점검해보자.

```

1 % This is a small sample LaTeX input file (Version of 10 April 1994)
2 %
3 % Use this file as a model for making your own LaTeX input file.
4 % Everything to the right of a % is a remark to you and is ignored by LaTeX.
5
6 % The Local Guide tells how to run LaTeX.
7
8 % WARNING! Do not type any of the following 10 characters except as directed:
9 %          & $ # % _ { } ^ ~ \
10
11 \documentclass{article}      % Your input file must contain these two lines
12 \begin{document}           % plus the \end{document} command at the end.
13
14
15 \section{Simple Text}       % This command makes a section title.
16
17 Words are separated by one or more spaces. Paragraphs are separated by
18 one or more blank lines. The output is not affected by adding extra
19 spaces or extra blank lines to the input file.
20
21 Double quotes are typed like this: ``quoted text''.
22 Single quotes are typed like this: `single-quoted text'.
23
24 Long dashes are typed as three dash characters---like this.
25
26 Emphasized text is typed like this: \emph{this is emphasized}.
27 Bold          text is typed like this: \textbf{this is bold}.
28
29 \subsection{A Warning or Two} % This command makes a subsection title.
30
31 If you get too much space after a mid-sentence period---abbreviations
32 like etc.\ are the common culprits)---then type a backslash followed by
33 a space after the period, as in this sentence.
34
35 Remember, don't type the 10 special characters (such as dollar sign and
36 backslash) except as directed! The following seven are printed by
37 typing a backslash in front of them: \$ \& \# \% \_ \{ and \}.
38 The manual tells how to make other symbols.
39
40 \end{document}              % The input file ends with this command.

```

그림 1: 샘플 L<sup>A</sup>T<sub>E</sub>X 파일

## 1 Simple Text

Words are separated by one or more spaces. Paragraphs are separated by one or more blank lines. The output is not affected by adding extra spaces or extra blank lines to the input file.

Double quotes are typed like this: “quoted text”. Single quotes are typed like this: ‘single-quoted text’.

Long dashes are typed as three dash characters—like this.

Emphasized text is typed like this: *this is emphasized*. Bold text is typed like this: **this is bold**.

### 1.1 A Warning or Two

If you get too much space after a mid-sentence period—abbreviations like etc. are the common culprits—then type a backslash followed by a space after the period, as in this sentence.

Remember, don’t type the 10 special characters (such as dollar sign and backslash) except as directed! The following seven are printed by typing a backslash in front of them: \$ & # % \_ { and }. The manual tells how to make other symbols.

---

그림 2: 샘플 파일의 처리 결과

---

## 1 텍스트

단어는 하나 이상의 스페이스로 구분한다. 문단은 하나 이상의 빈 줄로 구분한다. 스페이스나 빈 줄이 여러 개 추가되어도 출력에 영향을 미치지 않는다.

따옴표는 이런 모양이다: “따옴표”. 작은따옴표는 이런 모양이다: ‘작은따옴표’.

긴 대시는 세 개의 대시문자로 표시한다—이렇게.

강조를 하려면 강조된 단어와 같이 하라. 굵은 글자는 굵은 글자로 식자된다.

### 1.1 약간의 경고

온점이 마침표가 아니라 약어 기호로 쓰일 때와 같이 그 뒤에 따라붙는 공백이 너무 커서는 아니될 경우, 마침표 뒤에 백슬래시와 스페이스 하나를 주어라. 예를 들면, etc. 와 같이 한다.

열 개의 특별한 문자(달러화 기호나 백슬래시 등)는 텍스트로서 바로 입력하면 안 된다. 다음 몇 글자는 그 앞에 백슬래시를 붙여 얻을 수 있다: \$ & # % \_ { }. 그밖의 기호문자에 대해서는 사용설명서에서 소개한다.

---

그림 3: 한국어로 번역한 small2e.tex

### 3.1 본문(Running Text)

모든 문서는 거의 대부분이 평문단(running text)—단어로 이루어진 문장과, 그 문장으로 이루어진 단락들—으로 구성되어 있고, 이 점에 있어서는 예제 파일도 예외가 아니다. Running text를 기술하는 데는 아무런 문제가 없다. 일반적인 방법대로 입력하면 된다. 그에 따라 작성되는 출력물에서, L<sup>A</sup>T<sub>E</sub>X은 좌우측 여백을 말끔하게 정돈하기 위해 단어 간의 공백을 조정하고 행을 가득 채운다. 입력파일에서 공백을 주거나 단어를 띄워두는 것은 최종 출력물에 아무런 영향도 미치지 못한다. 입력파일에 들어 있는 공백이 몇 개이든간에 L<sup>A</sup>T<sub>E</sub>X은 이를 하나의 공백으로 처리한다. 또한 L<sup>A</sup>T<sub>E</sub>X은 각 행의 끝을 단어 간의 공백으로 간주한다(17–19행 참조). 입력파일에서 공백인 행은 새로운 단락의 시작임을 가리키므로, 새로운 단락을 시작하는 것이 아니라면 입력파일에 공백인 행을 두지 말아야 한다.

L<sup>A</sup>T<sub>E</sub>X은 잘 쓰이지 않는 키보드 문자를 특별하게 사용하기 위해 유보하고 있다. 사용자의 입력 파일에 이와 같은 열 개의 문자

# \$ % & ~ \_ ^ \ { }

를 사용해서는 안된다—만약 이를 사용한다면, L<sup>A</sup>T<sub>E</sub>X은 이를 제대로 처리하지 못하게 된다.

### 3.2 L<sup>A</sup>T<sub>E</sub>X 명령어

입력파일에는 ‘\’로 시작되는 단어들이 있다(11, 12 및 15행 참조). 이들은 L<sup>A</sup>T<sub>E</sub>X 명령어로서 사용자 문서의 구조를 기술하는 데 사용된다. 이러한 명령어의 특징은 다음과 같다.

- 모든 L<sup>A</sup>T<sub>E</sub>X 명령어는 ‘\’로 시작하고 하나 또는 그 이상의 문자로 이루어진다. 명령어는 빈칸(스페이스)이나 문자 아닌 것(숫자나 괄호 등)이 오기까지를 명령의 이름으로 인식한다.
- L<sup>A</sup>T<sub>E</sub>X 명령어는 대소문자를 구분한다. 따라서 \BEGIN과 \Begin 그리고 \begin은 각각 서로 다르다.
- 몇몇 명령어들은 사용자의 본문 중에 위치하기도 한다. 이러한 명령어들은 다른 글꼴과 같은 것을 선택하는 스위치와 같은 역할을 한다. `small12e.tex`에서는 본문의 강조—보통 이탤릭체 글꼴로 바꾸어 표현한다—를 위해 이를 사용하고 있다(26행 참조). 명령어와 본문은 항상 ‘{’와 ‘}’로 둘러싸여 있으며, 명령어 ‘{\em’은 기능을 활성화하고 ‘}’를 만나면 범위가 종료되어 주어진 기능을 비활성화한다. 이와 같은 글꼴 변경 명령어는 모두 인자를 취하는 명령어형을 별도로 가지고 있다. 자세한 것은 10페이지를 참조하라. 이와 같이 인자 없이 명령어 자체로서 특정한 기능을 설정하는 스위치 형 명령어를 ‘선언형’이라고 한다.
- 다음과 같은 형태의 명령어도 있다.

`\command{text}`

이 때, `text`를 명령어의 인자(parameter 또는 argument)라고 한다. 명령어 `\section`이 이와 같은 것이다(15행 참조). 인자는 활괄호 ‘{ }’나 각괄호 ‘[ ]’ 속에 표기한다. 경우에 따라서 두 가지 모두를 사용하기도 한다. 각괄호는 보통 없어도 되는 ‘선택 인자’이고

활괄호는 반드시 있어야 하는 ‘인자’이다. 인자를 가지는 명령어를 ‘명령형’ 명령이라고 한다. 명령어는 있는 그대로 정확하게 사용하여야 한다.

- 명령어는 오직 문자(letter)로만 이루어지기 때문에 문자가 아닌 것이 오면 명령어가 끝난다. 명령어의 끝을 식별하는 데 명령어의 인자를 담고 있는 활괄호나 공백을 사용하는 것이 일반적이다. 공백을 사용한 경우에 L<sup>A</sup>T<sub>E</sub>X은 그 공백을 무시한다. 이로 인하여 발생할 수 있는 문제에 대해서는 뒤에서 살펴보겠다.

### 3.3 전체적인 구조

모든 문서에 반드시 사용해야 하는 L<sup>A</sup>T<sub>E</sub>X 명령어가 있다. 실제 본문은 `\begin{document}` 명령어로 시작하고 `\end{document}` 명령어로 끝난다(12 및 40행 참조). `\end{document}` 이후에 있는 모든 것은 무시된다. `\begin{document}` 이전에 나타나는 것들을 전처리부(*preamble*)라고 한다. 전처리부(*preamble*)에는 문서의 형식을 기술하는 명령어만 사용하여야 한다.

전처리부(*preamble*)에서 반드시 사용하여야 하는 명령어가 `\documentclass`이다(11행 참조). 이 명령어는 문서의 전체적인 형식을 지정한다. 예제 파일은 간단한 기술문서이므로 `article` 형식을 사용하고, 11포인트 글꼴로 인쇄하도록 지정하였다. 4절에 나타나 있는 것과 같은 다른 형식의 사용도 가능하다.

### 3.4 기타 살펴볼 사항

L<sup>A</sup>T<sub>E</sub>X은 열고 닫는 홑따옴표와 겹따옴표를 모두 식자할 수 있다. 이는 키보드에 있는 두 개의 따옴표—` (grave accent 또는 back-quote와 유사하다)와 ' (apostrophe)—를 이용한 것이다. 홑따옴표에는 한 번(22행 참조), 겹따옴표에는 두 번을 사용한다(21행 참조). 키보드 겹따옴표 문자 자체 "는 거의 사용하지 않는다.

L<sup>A</sup>T<sub>E</sub>X은 서로 다른 세 가지의 대시를 사용한다. 문장에서 기호로 사용하는 긴 대시는 세 개의 대시 ‘---’를 연속해서 입력한다. 숫자의 범위 표시에 사용하는 짧은 대시 ‘10-20’는 두 개의 대시를 연속해서 입력하고 한 개의 대시는 하이픈으로 사용한다.

때때로 본문에 L<sup>A</sup>T<sub>E</sub>X에서 사용하는 특수 문자를 입력해야 할 필요가 있다. 그 중 일곱 개의 문자는 앞에서 백슬래시를 붙여 입력할 수 있다(37행 참조). 나머지 세 개의 문자는 `†`, `‡`, `§`, `£`, `©`, `‡`, `♣` 등 키보드에 없는 문자를 입력하는 것과 마찬가지로 별도의 명령어를 이용해 작성한다.

자신이 작업한 내용이나 이유 등을 알 수 있도록 L<sup>A</sup>T<sub>E</sub>X 파일에 주석을 첨가해 둔다면 매우 유용할 것이다. L<sup>A</sup>T<sub>E</sub>X은 % 기호 우측에 있는 모든 것을 무시하므로 이를 주석문으로 사용한다.

## 4 도큐먼트 클래스와 옵션

L<sup>A</sup>T<sub>E</sub>X에는 몇 가지 표준 도큐먼트 클래스가 있다.

`article` 짧은 문서나 출판물 속에 들어갈 기사와 같은 것에 어울린다. `article`은 `chapter`가 없고 `\maketitle`로 제목을 작성(9절 참조)하면, 표지를 별도로 사용하지 않고 첫 페이지의 상단에 제목을 인쇄한다.

`report` 좀더 기술적인 문서 작성에 알맞다. 이는 `chapter`를 가지며 제목을 별도의 페이지에 인쇄하는 점을 제외하면 `article`과 같다.

`book` 책자의 출판을 목적으로 한다. 종이의 양면에 인쇄할 것으로 가정하여 페이지의 구성을 조정한다.

`letter` 편지 등을 작성하기 위한 것이다. 편지 형식을 사용하면 주소, 날짜, 서명 등의 각 구성요소가 적절하게 배치된 편지를 작성할 수 있다.

`minimal` 폰트 크기와 용지를 제외하고는 아무 것도 사전설정하지 않는 클래스이다. 주로 디버깅 코드를 작성하기 위해서 사용한다.

`proc` proceeding 작성을 위한 `article`의 변형이다.

`slides` 큰 글자로 슬라이드를 제작하는 클래스이다.

이러한 표준 형식을 변경할 수 있는 옵션이 다수 존재한다. 명령어 `\documentclass` 다음의 각괄호 안에 들어 있는 것이 바로 그것이다. 문서에서 클래스는 오직 하나만이 사용 가능한 반면에, 옵션은 옵션명을 콤마로 구분하여 여러 개를 동시에 사용할 수 있다. 표준 옵션은 다음과 같다.

`11pt` 본문의 내용을 일반적으로 사용되는 10 포인트가 아닌 11 포인트 글꼴로 식자한다. 11 포인트는 10 포인트에 비해 약 10퍼센트 가량 크다.

`12pt` 본문의 내용을 일반적으로 사용되는 10 포인트가 아닌 12 포인트 글꼴로 식자한다. 12 포인트는 10 포인트에 비해 약 20퍼센트 가량 크다.

`twoside` `article`이나 `report` 형식에서 문서를 종이의 양면에 인쇄하도록 한다. 이는 `book` 형식에서 기본값이다.

`twocolumn` 각 페이지의 본문을 2단으로 조판한다.

`titlepage` `article` 형식에서 `\maketitle` 명령어가 새로운 페이지에 표지를 작성하도록 한다. `report`와 `book` 형식에서는 표지가 기본적으로 새로운 페이지에 작성된다.

유럽 국가에 유용한 옵션이 있다. 옵션 `a4paper`는 모든 표준 형식의 출력을 A4 용지에 맞게끔 조정한다. (L<sup>A</sup>T<sub>E</sub>X이 만들어진 미국에서는 A4 용지보다 길이가 조금 짧고 폭이 넓은 용지가 표준이어서, 이 옵션 없이 인쇄하면 어딘지 모르게 어색할 것이다.) 용지 설정을 위한 옵션은 이밖에도 몇 가지가 더 있다.

표준 도큐먼트 클래스에 줄 수 있는 여러 가지 옵션에 대해서 알고 싶다면 `LaTeX:Document Structure` 웹페이지를 참고하라.

## 5 환경

우리는 앞에서 L<sup>A</sup>T<sub>E</sub>X이 제대로 조판할 수 있도록 인용문을 식별하는 것에 대해 언급하였다. 인용문은 `\begin{quotation}`과 `\end{quotation}` 명령어로 둘러싸여 이것이 인용문임을 식별한다.

이것은 환경 (*environment*)이라고 불리는 L<sup>A</sup>T<sub>E</sub>X 구조의 한 예이다. 본문을 특정 환경에 배치하면 여러 가지 특수한 효과를 얻을 수 있다.

## 5.1 인용문 quote, quotation

인용에는 `quote`와 `quotation`의 두 가지 환경이 있다. `quote`는 짧은 인용문이나 공백인 행으로 분리된 짧은 인용문이 연속될 때 사용된다.

US presidents have been known for their pithy remarks:

The buck stops here.

I am not a crook.

```
US presidents have been
known for their pithy remarks:
\begin{quote}
The buck stops here.

I am not a crook.
\end{quote}
```

인용문이 한 단락을 넘어설 경우에는 `quotation` 환경을 사용한다. 일반적으로 입력에서 단락은 공백인 행으로 구분한다. `quotation`은 단락별로 들여쓰기가 된다.

Here is some advice to remember:

Environments for making quotations can be used for other things as well.

Many problems can be solved by novel applications of existing environments.

```
Here is some advice to remember:
\begin{quotation}
Environments for making quotations
can be used for other things as well.

Many problems can be solved by
novel applications of existing
environments.
\end{quotation}
```

운문(시)을 조판하는 `verse` 환경이 있다. 각 행은 `\\`로 구분하고 연은 한 줄을 띄어쓴다. 행이 길면 줄을 나누는데 같은 행에 속하는 나누어진 줄은 들여밀기 식자된다.

I will arise and go now, and go to  
Innisfree,

And a small cabin build there, of  
clay and wattles made:

Nine bean-rows will I have there,  
a hive for the honey-bee;

And live alone in the bee-loud  
glade.

And I shall have some peace  
there, for peace comes  
dropping slow,

Dropping from the veils of the  
morning to where the  
cricket sings;

There midnight's all a glimmer,  
and noon a purple glow,

And evening full of the linnet's  
wings.

```
\begin{verse}
I will arise and go now,
and go to Innisfree,\\
And a small cabin build there,
of clay and wattles made:\\
Nine bean-rows will I have there,
a hive for the honey-bee;\\
And live alone in the bee-loud glade.

And I shall have some peace there,
for peace comes dropping slow, \\
Dropping from the veils of the morning
to where the cricket sings; \\
There midnight's all a glimmer,
and noon a purple glow, \\
And evening full of the linnet's wings.
\end{verse}
```



## 5.2 중앙 및 좌우정렬, center, flushright, flushleft

center 환경에 배치된 본문은 페이지의 중앙에 정렬되고, flushleft나 flushright 환경에 배치된 본문은 좌·우측 끝에 정렬된다. center의 철자에 주의하라—불행히도 L<sup>A</sup>T<sub>E</sub>X은 영국식 철자 (centre)를 받아들이지 못한다.

이와 같은 환경 속에 있는 본문은 일반적인 방식대로 조판된다. 특히 사용자가 입력한 행의 끝 문자는 단순히 공백으로 간주된다. 강제로 새로운 행을 시작하려면 \\ 명령어를 입력하여야 한다. 예를 들어보자.

```
one two three
      four
      five
```

```
\begin{center}
one
two
three \\
four \\
five
\end{center}
```

## 5.3 나열문단, itemize, enumerate, description

리스트를 작성하기 위한 세 가지 환경이 존재한다. 각각의 환경에서 새로운 항목은 \item 명령어로 시작한다. enumerate 환경에서는 각 항목이 번호로 표시되는 반면에 itemize 환경에서는 특정한 마크로 표시된다. 이 환경들은 서로 내포될 수 있으며, 이 경우 들여쓰기의 양과 마크는 적절히 조정된다.

- Itemized lists are handy.
- However, don't forget
  1. The 'item' command.
  2. The 'end' command.

```
\begin{itemize}
\item Itemized lists are handy.
\item However, don't forget
  \begin{enumerate}
  \item The 'item' command.
  \item The 'end' command.
  \end{enumerate}
\end{itemize}
```

설명형 리스트 작성을 위한 환경이 description이다. description 환경에서는 \item 명령어 뒤의 각괄호에 항목의 레이블을 지정할 수 있다. 예를 들어보자.

Three animals that you should know about are:

**gnat** A small animal that causes no end of trouble.

**gnu** A large animal that causes no end of trouble.

**armadillo** A medium-sized animal.

```
Three animals that you should
know about are:
\begin{description}
\item[gnat] A small animal that causes
no end of trouble.
\item[gnu] A large animal that causes
no end of trouble.
\item[armadillo] A medium-sized animal.
\end{description}
```

## 5.4 그대로 보이기, verbatim

경우에 따라서는 터미널 화면에 나타난 모양대로 본문을 입력할 필요가 있다. 컴퓨터 프로그램의 소스코드와 같은 것이 그 예가 될 것이다. 사용자가 원하는 것은 L<sup>A</sup>T<sub>E</sub>X이 사용자의 본문 배치를 중단하는 것만이 아니라 L<sup>A</sup>T<sub>E</sub>X에게 아무런 영향도 주지 않고 키보드에 있는 모든 문자들을 사용하는 것도 포함될 것이다. `verbatim` 환경이 이러한 기능을 제공한다.

The section of program in question is:

```
{ this finds %a & %b }
```

```
for i := 1 to 27 do
  begin
    table[i] := fn(i);
    process(i)
  end;
```

The section of program in question is:

```
\begin{verbatim}
{ this finds %a & %b }
```

```
for i := 1 to 27 do
  begin
    table[i] := fn(i);
    process(i)
  end;
\end{verbatim}
```

`verbatim` 환경은 문단을 그대로보이기하는 데 사용한다. 한편, 한두 글자 또는 단어에 대하여 그대로보이기하려면 `\verb`라는 명령어를 쓴다. 이 명령은 일반적으로 인자를 취하는 명령어와 달리 시작문자와 끝문자를 표시하는 방법으로 써야 한다. 예를 들어 `\verb+&#+`에서와 같이 + 문자를 시작과 끝문자로 사용하였다. 다른 문자, 예를 들면 |를 쓰면 `\verb|&#|`와 같이 할 수 있다.

별표붙인 환경 `verbatim*`이나 `\verb*`는 빈 칸(스페이스)에 특별한 기호를 써서 표시한다. 다음 보기를 참고하라.

```
for_i_i:=_1_to_27_do
```

따라서 \* 문자는 `\verb`의 시작 끝 문자로 사용하면 안 된다.

## 6 글꼴

우리는 이미 강조 글꼴을 사용하기 위해 `\emph` 명령을 사용했었다. 글꼴을 변경하기 위한 명령어의 전체 리스트가 여기에 있다.

<i>Command</i>	<i>or</i>	<i>Effect</i>
<code>\textrm{...}</code>	<code>{\rmfamily...}</code>	Text is set in roman family
<code>\textsf{...}</code>	<code>{\sffamily...}</code>	Text is set in sans serif family
<code>\texttt{...}</code>	<code>{\ttfamily...}</code>	Text is set in typewriter family
<code>\textmd{...}</code>	<code>{\mdseries...}</code>	Text is set in medium series
<code>\textbf{...}</code>	<code>{\bfseries...}</code>	Text is set in <b>bold</b> series
<code>\textup{...}</code>	<code>{\upshape...}</code>	Text is set in upright shape
<code>\textit{...}</code>	<code>{\itshape...}</code>	Text is set in <i>italic</i> shape
<code>\textsl{...}</code>	<code>{\slshape...}</code>	Text is set in <i>slanted</i> shape
<code>\textsc{...}</code>	<code>{\scshape...}</code>	Text is set in SMALL CAPS shape

<code>\emph{...}</code>	<code>{\em ...}</code>	Text is set <i>emphasized</i>
<code>\textnormal{...}</code>	<code>{\normalfont...}</code>	Text is set in the document font

두번째 열에 있는 선언형은 그 효력이 미치는 범위를 제한하기 위하여 한 쌍의 괄호 안에서 사용된다. 다른 모양으로 조판하고자 하는 텍스트를 인자로 하는 명령형(첫번째 열)의 사용을 권한다. 글꼴 명령에 더하여, 글꼴의 크기를 변경하는 명령어도 있다.

`tiny` `scriptsize` `footnotesize` `small` `normalsize`

`large` `Large` `LARGE` `huge`

`Huge`

```

\tiny tiny
\scriptsize scriptsize
\footnotesize footnotesize
\small small
\normalsize normalsize
\large large
\Large Large
\LARGE LARGE
\huge huge
\Huge Huge
    
```

## 7 섹션 명령과 목차

학술논문이나 기술문서는 보통 여러 섹션으로 구분한다. 각각의 섹션은 참조를 보다 손쉽게 하기 위해서 제목과 번호를 부여한다. L<sup>A</sup>T<sub>E</sub>X에는 각기 다른 종류의 섹션을 식별하기 위한 일련의 명령어가 준비되어 있다. 일단 사용자가 이러한 명령어를 사용하면, 제목의 배치와 번호의 부여는 L<sup>A</sup>T<sub>E</sub>X이 처리한다.

이러한 명령어에는 다음과 같은 것들이 있다.

<code>\chapter</code>	<code>\subsection</code>	<code>\paragraph</code>
<code>\section</code>	<code>\subsubsection</code>	<code>\subparagraph</code>

뒤의 두 명령어는 명칭이 적절하지 못하다. 일반적인 단어의 의미대로 단락과 관련된 것이 아니라 이는 단지 section의 하위 레벨일 뿐이다. 대부분의 문서에서 `\paragraph`와 `\subparagraph`에서 만들어진 제목에는 번호를 부여하지 않는다. `article` 형식에서는 `\chapter`를 사용할 수 없다. 각 명령어는 section은 chapter 내에서, subsection은 section 내에서 사용하는 것과 같이 주어진 순서대로 호출하여야 한다.

일곱번째 섹션 명령인 `\part`도 있다. 이 명령어의 사용은 선택적으로서, 이는 대용량의 문서를 일련의 부분으로 구분하는 데 사용된다. `\part`가 사용되어도 `\chapter`의 번호가 갱신되지 않는다.

사용자는 문서에서 `\tableofcontents`를 사용하여 각종 섹션 명령을 바탕으로 하여 작성된 내용에 따른 목차를 작성·포함할 수 있다. 문서를 매번 L<sup>A</sup>T<sub>E</sub>X으로 처리할 때마다 직전에 L<sup>A</sup>T<sub>E</sub>X 처리에서 얻어진 제목 등으로 목차가 새롭게 구성된다는 점에 유의하여야 한다. 이는 L<sup>A</sup>T<sub>E</sub>X 문서를 처리할 때 목차에 필요한 정보를 모아두었다가 다시 문서를 처리할 때 이 정보를 이용하기 때문이다. 이 때문에 정확한 목차를 작성하기 위해서 문서를 L<sup>A</sup>T<sub>E</sub>X으로 두 번 이상 처리하여야 할 경우도 있다.

## 8 특수기호의 식자

사용자는 키보드에 없는 다양한 종류의 특수기호들을 자신의 문서에 나타낼 수 있다. 먼저 어떠한 글자에도 액센트를 표시할 수 있다.

```

ò \`{o}   õ \~{o}   ö \v{o}   o \c{o}   ó \'{o}
ō \={o}   ö \H{o}   ȝ \d{o}   ô \^{}{o}   ô \.{}{o}
oo \t{oo}  ȝ \b{o}
ö \"{o}   ȝ \u{o}

```

기타의 특수기호와 그 입력 명령어를 아래에 표시하였다.

†	<code>\dag</code>	§	<code>\S</code>	©	<code>\copyright</code>
‡	<code>\ddag</code>	¶	<code>\P</code>	£	<code>\pounds</code>
œ	<code>\oe</code>	Œ	<code>\OE</code>	æ	<code>\AE</code>
Æ	<code>\AE</code>	å	<code>\aa</code>	Å	<code>\AA</code>
ø	<code>\o</code>	Ø	<code>\O</code>	ł	<code>\l</code>
Ł	<code>\E</code>	ß	<code>\ss</code>	?’	<code>?`</code>
!‘	<code>!`</code>	…	<code>\ldots</code>	L <sup>A</sup> T <sub>E</sub> X	L <sup>A</sup> T <sub>E</sub> X

유보문자 가운데 `\`를 붙여서 얻을 수 없는 것이 `\`이다. `\`는 행을 강제로 나누는 데 쓰이는 것으로서 `\` 문자를 표시하지 않는다. 다음은 유보문자를 텍스트로 식자하는 명령어이다.

```
\textbackslash, \textasciicircum, \textasciitilde
```

이 입력의 결과는 다음과 같다: `\^~`

명령어 `\today`는 현재 날짜를 문서에 삽입한다.

이러한 L<sup>A</sup>T<sub>E</sub>X 명령어를 사용할 때에는 L<sup>A</sup>T<sub>E</sub>X이 명령어 뒤의 모든 공백을 무시한다는 점에 유의해야 한다. 따라서 `\pounds 20`라고 입력하면 ‘£20’과 같이 나타난다. `\LaTeX is wonderful`라고 입력하면 ‘L<sup>A</sup>T<sub>E</sub>X is wonderful’이라고 나타나는데, 출력 결과의 L<sup>A</sup>T<sub>E</sub>X 다음의 공백이 없다는 것에 주목하라. 이를 방지하기 위해서는 명령어 다음에 한 쌍의 빈 중괄호를 입력한 뒤에 공백을 입력하면 된다. 이제 `\LaTeX{} really is wonderful!`은 ‘L<sup>A</sup>T<sub>E</sub>X really is wonderful!’과 같이 나타날 것이다.

[역자의 추가] 한국어와 같이 조사가 있는 언어에서 명령어와 조사를 붙여쓰는 문제가 있다. 한글도 문자(letter)로 간주되는 T<sub>E</sub>X 엔진에서 예를 들어 ‘L<sup>A</sup>T<sub>E</sub>X은’과 같이 입력하면 이런 명령어 정의되지 않았다고 할 수 있다. 그러므로 조사를 그에 앞선 명령어로부터 떨어뜨려두는 것이 좋다. ‘L<sup>A</sup>T<sub>E</sub>X 은:’ 여기서 공백은 스페이스가 아니라 앞의 명령어가 끝났음을 의미하는 표지이다. 이런 일은 현재까지의 한국어/한글 처리 패키지에서는 발생하지 않는 문제이기도 하다. 예를 들어 ko.T<sub>E</sub>X은 이런 문제를 피하기 위해서 한글을 일반 문자가 아닌 것처럼 예외적으로 처리한다.

마지막으로 보통 수식의 조판에 사용하는  $\LaTeX$ 의 수식 모드에서는, 대·소문자의 희랍어 수학 알파벳, 멋진 수식 글꼴, 수학 연산자와 관계 연산자 및 화살표, 기타 훨씬 더 많은 기호를 사용할 수 있다. 이에 관해서는 13절에서 자세히 다룬다.

## 9 표제

대부분의 문서에는 제목과 저자가 표시되는 표제가 있다.  $\LaTeX$  문서에 표제를 만들기 위해서는, 보통 `\begin{document}` 명령어 뒤에 다음의 명령어를 사용한다.

```
\title{required title}
\author{required author}
\date{required date}
\maketitle
```

문서의 작성자가 여럿일 경우에는 `\author` 명령어에 각 작성자의 이름을 `\and`로 구분하여 표시하면 각자의 이름이 서로 다른 행에 중앙정렬되어 인쇄된다. `\date` 명령어를 사용하지 않으면 현재의 날짜가 인쇄된다.

```
Essential  $\LaTeX$ 

J Warbrick A N Other

14th February 1988
```

```
\title{Essential  $\LaTeX$ }
\author{J Warbrick \and A N Other}
\maketitle
```

표제의 실제 모양은 도큐먼트 클래스를 따른다. `report`와 `book` 형식에서는 표제가 별면의 표지에 인쇄되고, `article` 형식에서는 첫번째 페이지의 상단에 인쇄된다. 클래스 옵션인 `titlepage`를 이용하면 이를 변경할 수 있다(4절 참조).

## 10 표의 사용

$\LaTeX$ 은 연속된 공백을 하나의 공백으로 처리하기 때문에 표를 조판하기가 까다로운 편이다. 어떻게 나타나는지 예제를 보자.

```
Income Expenditure Result
20s 0d 19s 11d happiness
20s 0d 20s 1d misery
```

```
\begin{flushleft}
Income Expenditure Result \\\
20s 0d 19s 11d happiness \\\
20s 0d 20s 1d misery \\\
\end{flushleft}
```

`tabbing` 환경을 이용하면 이 문제를 해결할 수 있다. `tabbing` 환경에서는 타이프라이터에서와 같이 탭스톱을 설정하여 탭으로 이동할 수 있다. 탭스톱은 `\=` 명령으로 설정하고, `\>` 명령은 다음 탭으로 이동한다. 줄바꿈은 `\\` 명령으로 이루어진다. `\kill`로 표기한 부분은 아무것도 출력되지 않으므로 탭스톱의 설정에 이용된다.

```
Income Expenditure Result
20s 0d 19s 11d Happiness
20s 0d 20s 1d Misery
```

```
\begin{tabbing}
Income \=Expenditure \= \kill
Income \>Expenditure \>Result \\\
20s 0d \>19s 11d \>Happiness \\\
20s 0d \>20s 1d \>Misery \\\
\end{tabbing}
```

타이프라이터에서의 탭키와는 달리, `\>` 명령은 비록 그 위치가 왼쪽이라 해도 순서대로 다음 탭으로 이동한다. 이 때문에 두 탭스톱 간의 간격이 너무 적을 경우에는 글자가 겹쳐서 써지는 현상이 나타난다.

좀더 복잡한 표작업을 한다면, `tabular` 환경이 필요할 것이다. 여기서는 L<sup>A</sup>T<sub>E</sub>X이 전체 표를 검토하여 가장 긴 항목을 집어넣기 위해서는 열의 간격을 어떻게 해야 할지를 결정하기 때문에 `tabbing`에서와 같이 열의 너비에 대해 염려하지 않아도 된다. 사용자는 단지 몇 개의 열이 있고 각 열을 어떻게 배치하고 싶은지를 L<sup>A</sup>T<sub>E</sub>X에게 알려주기만 하면 된다. 이는 `\begin{tabular}` 명령 다음에 각 열에 대해 한 문자로 된 *template*을 이용하여 지정한다.

- l 컬럼의 모든 셀 내용을 왼쪽정렬
- r 컬럼의 모든 셀 내용을 오른쪽정렬
- c 컬럼의 모든 셀을 가운데정렬

한 행 내에서의 항목 구분은 `&` 문자로 이루어지며(이 글자를 왜 특별히 취급해야 하는지 이제 알 수 있을 것이다), 한 행의 끝은 `\\`로 표시된다. 3열 2행으로 된 간단한 표를 작성해보면 다음과 같다.

<i>Name</i>	<i>Age</i>	<i>Height</i>
Sebastian	45	195cm
Mathew	1	68cm

```
\begin{tabular}{lrr}
\em Name & \em Age & \em Height \\
Sebastian & 45 & 195cm \\
Mathew & 1 & 68cm \\
\end{tabular}
```

좀 더 멋있게 조판할 수 있도록 도움을 주는 많은 보조적인 기능들이 있다.

1. `\hline`을 사용하면 표의 행과 행 사이에 줄을 그을 수 있다.
2. `template`에 `l`를 사용하면 열과 열 사이에 선을 그을 수 있다.
3. `template`에 `p`를 사용하고 `{`와 `}` 사이에 폭을 지정함으로써 열의 폭을 고정시킬 수 있다. 이때 폭은 'cm', 'mm' 또는 'in'으로 표기할 수 있다('inches'를 사용할 수는 없다).

좀더 세밀한 표를 작성한 예를 그림 4에서 보였다.

엔터를 입력하는 것이 중요한 것이 아니다. 행의 끝은 소스의 캐리지 리턴에 의해서가 아니라 `\\`로 이루어진다.

## 11 표와 그림

표는 페이지 사이에서 잘라질 수 없다. 항상 한 페이지 안에 가두어 배치되어야 한다. (예외적으로 여러 페이지에 걸치는 표가 있을 수 있지만 여기서는 논외로 한다.) 표를 만들어야 하는 위치에서 페이지 아랫부분이 충분히 남지 않았을 때는 어떻게 되는가?

사진이나 그림이 페이지 사이에서 잘라질 수 없는 것은 당연하다. 이와 마찬가지로 문제를 사진이나 그림을 넣을 공간을 어떻게 확보할 것인가?

사용자가 본문을 입력하는 동안은 L<sup>A</sup>T<sub>E</sub>X이 어디에서 페이지를 시작할지 알 수 없다. 그러므로 현재 페이지의 남은 공간이 얼마나 될지도 전혀 알 수 없다. L<sup>A</sup>T<sub>E</sub>X은 정확히 사용자의 입력

<i>Group</i>	<i>Type</i>	Sherds
Groups 1–9	Grey wares	219
Groups 40–44	Black (mostly 'black burnished') ware	116
Groups 61–67	Buff-red-orange wares	46
Groups 81–85	Colour-coated fine wares	67
Groups 91–2, 93–4	Mortaria and mis- cellaneous	35
Group 96	Samian	56
		538

```

\begin{tabular}{|l|p{1in}|r|} \hline
\em Group &\em Type &Sherds \\ \hline
Groups 1--9 & Grey wares & 219 \\ \hline
Groups 40--44 & Black (mostly `black
burnished') ware & 116 \\ \hline
Groups 61--67 & Buff-red-orange wares
& 46 \\ \hline
Groups 81--85 & Colour-coated fine
wares & 67 \\ \hline
Groups 91--2, 93--4 & Mortaria and
miscellaneous & 35 \\ \hline
Group 96 & Samian & 56 \\ \hline
&& 538 \\ \hline
\end{tabular}

```

그림 4: 표 작성 보기

파일에 있는 바로 그 위치가 아니라, 페이지의 적당한 곳에다 스스로 표나 그림을 가져다두는 방식으로 이 문제를 해결한다. 이와 같은 것을 ‘떠다니는 개체 (*floating objects*)’라고 한다. 현재 페이지의 상단이나 하단, 만약 여유가 충분하다면 현재 위치, 또는 아예 그림과 표로만 이루어지는 별도의 페이지를 선정하여 그 위치에 그림과 표를 배치한다. *table*과 *figure*가 대표적인 떠다니는 개체이다. 떠다니는 개체에는 캡션을 달 수 있다. (나중에 그림을 오려붙일 공간으로) 3인치의 공간을 확보하고 캡션을 단 예를 들어본다.

```

\begin{figure}
\vspace{3in}
\caption{A Photograph of my Subject}
\end{figure}

```

표와 그림의 번호는 섹션에서와 같이 자동으로 붙여진다. `\tableofcontents` 명령이 각 섹션과 서브섹션 등의 목록을 페이지 번호와 함께 표시해주듯이, `\listoffigures`와 `\listoftables` 명령은 그림과 표의 캡션을 모아서 목록을 만든다.

## 12 교차참조와 인용

### 12.1 교차참조

LaTeX의 가장 유용한 기능 중의 하나가 수식, 표, 그림, 페이지, 섹션 등의 번호를 나중에 참조할 수 있도록 해주는 것이다. 예를 들어, “도큐먼트 스타일에 대해서는 4절을 참조”하고, “`sample.tex`의 처리 결과는 그림 2를 참조”하는 것인데 다음과 같이 입력한다.

```

도큐먼트 스타일에 대해서는 제~\ref{sec:styles}~절을 참조하고,
\fn{sample.tex}의 처리 결과는 그림~\ref{fig:result}\를 참조

```

문서에서 참조번호가 인쇄된 곳에 직접 번호를 입력하지 않고 `\ref` 명령어를 썼음을 알 수 있다. ~는 줄바꿈을 할 수 없는 공백을 삽입한다.

교차참조를 하기 위해서는 L<sup>A</sup>T<sub>E</sub>X 입력 파일에서 참조될 부분도 당연히 표시되어야 한다. 이는 `\label` 명령으로 이루어진다. 섹션의 레이블은 다음처럼 섹션 제목 바로 뒤에 지정한다.

```
\section{Document Classes and Options}
\label{sec:styles}
```

그리고 figure나 table의 경우에는 `\caption` 명령 바로 다음에 이를 지정한다.

```
\caption{The Result of Processing the Sample File}
\label{fig:result}
```

`\label`과 `\ref`에는 어떠한 이름일지라도 사용할 수는 있지만, 위의 관행을 따르는 것이 나쁘지 않다. `\label`은 페이지 참조에도 사용된다. 다만 `\ref`가 아니라 `\pageref`를 사용한다.

## 12.2 문헌목록 데이터베이스를 이용하는 문헌목록과 인용

L<sup>A</sup>T<sub>E</sub>X 입력 파일 내에서 문서의 일부분을 기호로 참조하는 것과 동일하게, 다른 문서를 인용할 수도 있다. 문서에 대한 인용은 다음처럼 `\cite` 명령을 사용해서 이루어진다.

```
The book by Lamport \cite{Lamport-LaTeX} is the principal
reference work on \LaTeX.
```

보통 *citation key*라고 하는 `\cite` 명령어의 인자는 *bibliographic database*에 들어 있는 책자나 논문을 구분·인식하는 데 이용된다.

문서에서 참고문헌을 인용한다면, 다음의 명령어로 참조문헌 목록을 만들 수 있다.

```
\bibliographystyle{plain}
\bibliography{mybib1,mybib2}
```

L<sup>A</sup>T<sub>E</sub>X에서 인용의 형식은 `\bibliographystyle` 명령에 따른다. 표준적인 모양은 다음과 같다.

`plain` 인용물의 목록은 알파벳 순으로 정렬되어 번호가 부여되고, 문서 내에서는 각괄호 안에 표기된다.

`unsrt` 인용물의 목록은 문서에서 나타나는 순서대로 정렬되어 번호가 부여되고, 문서 내에서는 각괄호 안에 표기된다.

`alpha` 인용물의 목록은 알파벳 순으로 정렬되나 번호가 아니라 “Lam86”과 같이 레이블이 붙여지고, 문서 내에서는 각괄호 안에 이 레이블이 표기된다.

`abbrev alpha`와 비슷하나 전체적으로 더 간단하다. (예: 잡지의 이름과 월을 줄이는 등)

명령어 `\bibliography`의 인자는—위의 예제에서 `mybib1.bib`나 `mybib2.bib`와 같이—확장자가 `.bib`인 파일 이름을 콤마로 구분하여 표기한다. 이 파일에는 `\cite` 명령에 대한 완전한 참조사항이 들어 있다. 이와 같이 `.bib` 참고문헌 데이터베이스를 이용하여 문헌목록을 관리할 때는 `bibtex`이라는 유틸리티를 필요로 한다. `bibtex`의 사용법, `.bib` 파일의 형식, 인용을 L<sup>A</sup>T<sub>E</sub>X과 호환성있는 형식으로 변환하는 방법 등의 문제는 이 안내서의 범위를 넘어서는 것이다.



## 12.3 간이 문헌목록과 인용

이 문서에는 이와 같은 문헌목록 데이터베이스를 이용하지 않고 간단한 간이 문헌목록을 만드는 방법을 사용하였다 (17절). 이 문서의 문헌목록은 다음과 같이 작성되어 있다.

```
\begin{thebibliography}{9}
\bibitem{Lampport85} Leslie Lampport, \textit{\LaTeX---A Document
  Preparation System---User's Guide and Reference
  Manual}. Addison-Wesley, Reading, MA, USA, 1985.
\bibitem{Knuth86} Donald Knuth, \textit{The \TeX{}}book}, volume A of
  \textit{Computers and Typesetting}. Addison-Wesley, Reading, MA,
  USA, 1986.
\bibitem{companion} Michel Goossens, Frank Mittelbach and Alexander
  Samarin, \textit{The \LaTeX{}} Companion}. Addison-Wesley, Reading,
  MA, USA, 1993.
\end{thebibliography}
```

여기서는 `thebibliography` 환경을 사용하였고 하나의 인자를 취한다. 이 인자에는 어떤 것이든 올 수 있는데 중요한 것은 거기에 사용된 문자의 길이이다. 위의 예에서 {9}라고 한 것은 문헌목록의 개수가 한 자릿수라는 의미이다. 열 개가 넘는다면 {99}와 같이 입력한다.

`\bibitem`의 첫 인자는 citation key이다. 예를 들어 `Lampport85`로 식별되는 문헌은 본문에서 `\cite{Lampport85}`로 인용할 수 있다 [1].

## 13 수식의 조판

### 13.1 Math, Display-math 그리고 Equation

$\TeX$ 은 수식을 일반적인 텍스트와는 완전히 달리 취급한다. 이를 `math mode`라고 한다. `math mode`는 행중 (in-line)의 수식 표현을 위한 `math mode`와 별행으로 수식을 식자하는 `display math mode` 두 가지의 출력형식이 존재한다.

행중 수식은 `\(...\)`나 `$...$`로 둘러싸서 표기한다.

Some mathematics set inline  $2 \times 3 = 6$ . Note that spaces in the input file are ignored in math mode.

Some mathematics set inline  
`\( 2\times 3 = 6 \)`.  
Note that spaces in the input file  
are ignored in math mode.

별행 수식 (`display math`)는 `\[...\]`로 둘러싸서 표기한다.

A larger equation to be displayed on a line by itself.

$$f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x)}{i!}$$

A larger equation to be displayed on  
a line by itself.  
`\[ f(x) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x)}{i!} \]`

별행 수식의 일종인 `equation`은 자동적으로 수식에 번호를 붙인다.

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} = \begin{pmatrix} 4 & 6 \\ 1 & 3 \end{pmatrix} \quad (1)$$

```
\begin{equation}
\left(\begin{array}{cc}
1 & 2 \\
0 & 1
\end{array}\right)\left(\begin{array}{cc}
2 & 0 \\
1 & 3
\end{array}\right) =
\left(\begin{array}{cc}
4 & 6 \\
1 & 3
\end{array}\right)
\end{equation}
```

이 예제에서는 math mode에서 사용하는 주요 명령어를 보여주고 있다. 다음과 같은 점에 주목하여야 한다.

1. 지수와 첨자는 `_`와 `^`로 표기한다. 예를 들어  $x_{\{1\}} = p^{\{2\}}$ 는  $x_1 = p^2$ 와 같이 나타난다.
2. 분수는 `\frac` 명령으로 표시한다. 예를 들어  $\frac{a+b}{c}$ 는  $\frac{a+b}{c}$ 로 나타난다.
3. 수학기호를 나타내는 많은 명령이 있다. `\infty` `\rightarrow` `\surd` `\bigotimes`는  $\infty \Rightarrow \sqrt{\otimes}$ 와 같이 표시된다.
4. 행렬을 `array` 환경으로 표현하였다. 이는 각 항목이 텍스트모드가 아니라 math mode로 조판된다는 점을 제외하면 10절에서 설명한 `tabular` 환경과 동일하다. `array` 환경에서는 배열에 괄호를 표시하지 않는 데 주목하여야 한다. 따라서 행렬이나 심지어는 열을 중심으로 정렬하고자 하는 일련의 수식에도 이를 이용할 수 있다.
5. 명령어 `\left`와 `\right`를 사용하면 필요한 짝맞춤 문자를 적절한 크기로 조판할 수 있다. 짝맞춤 문자로 사용할 수 있는 많은 기호가 있다. e.g., `\left(` `\left\{` `\left|`. 구분자 전체 목록을 아래의 표 A.8과 표 A.9에 나타내었다.

## 13.2 간격

Math mode에서는 입력파일에 있는 모든 간격을 무시한다. 간격을 조정하고자 할 때에는 다음의 명령을 사용한다.

`\,` thin space                    `\:` medium space  
`\!` negative thin space        `\;` thick space

간격의 수동 조정이 필요한 좋은 예가 있다.

$$\iint z \, dx \, dy. \iint z \, dx \, dy$$

```
\[
\int\!\!\int z \, dx \, dy ..
\int\int z \, dx \, dy
\]
```

### 13.3 수식 모드에서 글꼴의 변경

수식 모드의 기본 글꼴은 *math italic*이다. 이를 일반적인 *text italic*과 혼동해서는 안된다. 일반적인 문자의 글꼴은 일반적인 명령인 `\emph`, `\textbf`, etc. 등으로 변경할 수 있다. 희랍어 소문자(`\alpha` 등)는 수학기호로 간주되고(이는 수식 모드에서 입력해야 한다는 것을 의미한다), 이러한 글꼴 변경 명령에 영향을 받지 않는다.

명령어 `\mathbf`는 **bold face roman** 문자를 식자한다. 볼드체 희랍어나 수학기호 및 **bold face math italic**을 사용하고자 한다면 수식 모드에 들어가기 전에 `\boldmath` 명령을 사용하라. 그러면 수식 모드의 기본 글꼴이 볼드로 변경된다.

$x = 2\pi \Rightarrow x \simeq 6.28$   
 $\mathbf{x} = 2\pi \Rightarrow x \simeq 6.28$   
 $\mathbf{x} = \mathbf{2\pi} \Rightarrow \mathbf{x} \simeq \mathbf{6.28}$

```
\( x = 2\pi \Rightarrow x \simeq 6.28 \)

\(\ \mathbf{x} = 2\pi \Rightarrow x \simeq 6.28 \)

{\boldmath
\(\mathbf{x} = \mathbf{2\pi} \Rightarrow \mathbf{x} \simeq \mathbf{6.28}\)}
\}
```

대문자에 대해서 calligraphic 글꼴이 존재한다. 이는 `\mathcal` 명령으로 생성한다.

$\mathcal{F}$

```
\(\ \mathcal{F} \)
```

### 13.4 \$가 의미하는 것은?

*AMS-TeX*이나 plain *TeX*을 사용하다가 *LaTeX*으로 전환했다면, 왜 \$나 \$\$에 대해 언급이 없는지 궁금할 것이다.

여기서는 *math mode*를 \$로 둘러싸서 표시하고, *display math mode*는 \$\$로 둘러싸서 표시한다. 이렇게 하면 문서 편집기에서 \$ 기호의 짝을 맞추기가 어렵고 *math mode*의 시작과 끝이 어딘지를 알기가 어려우므로 *LaTeX* 시스템에 비해 단점이 된다. 사용자가 \$ 기호를 빠뜨리면 *TeX*도 혼란스럽게 된다.

(잘못된) 입력

```
let (a,b,c)$ be a Pythagoream triple, i.e.\ three
integers such that $a^{2}+b^{2}=c^{2}$.
```

은 다음과 같은 약간 이상한 에러 메시지를 나타낸다.

```
! Missing $ inserted.
<inserted text>
$
<to be read again>
```

```
1.56 ...triple, i.e.\ three integers such that $a^{
2}+b^{2}=c^{2}$
?
```

위의 예에서 수식 모드 바깥에서 `~` 명령을 사용하였다고 하여 에러의 내용과 위치를 잘못 표기하고 있다. T<sub>E</sub>X은 ‘be a ... such that’을 수식 모드로 조판하고 ‘such that’ 다음의 \$에서 수식 모드를 마친다.

동일한 실수를 L<sup>A</sup>T<sub>E</sub>X에서 했다면, 사용자의 의도를 좀더 정확히 짐작한다.

```
let (a,b,c)\) be a Pythagorean triple, i.e.\ three
integers such that \((a^{2}+b^{2}=c^{2})\)
```

에러 메시지의 내용은 여전히 명칭하지만, (a,b,c)의 앞에서 \((를 빠뜨렸으므로 수식 모드가 아닌 곳에서 수식 모드의 끝을 표시하는 \))를 사용하였다고 정확한 위치를 알려준다.

```
LaTeX error. See LaTeX manual for explanation.
Type H <return> for immediate help.
! Bad math environment delimiter.
\@latexerr ...for immediate help.}\errmessage {#1}
```

```
\)...ifinner $\else \@badmath \fi \else \@badmath
\fi
```

```
1.56 let (a,b,c)\)
be a Pythagorean triple, i.e.\ three integers such \...
?
```

달라 기호는 짧은 수식의 표현에서 유용하다.

Let  $G$  be a  $p$ -group

Let  $G$  be a  $p$ -group

두 개의 달라 기호가 항상 `[ ... ]`와 동일한 것은 아니므로, 다른 스타일이나 옵션에서도 호환성을 유지하고 싶다면 이를 사용하지 말아야 한다. (*fleqn* 클래스 옵션을 사용해 보라).

### 13.5 기호

부록에 표 A.1에서 표 A.19까지 L<sup>A</sup>T<sub>E</sub>X의 표준 기호를 종류별로 정리해두었다. 표 A.12에서 표 A.18에 있는  $\mathcal{A}\mathcal{M}\mathcal{S}$ -T<sub>E</sub>X의 추가적인 기호들은 `amssymb` 패키지를 지정해야지만 사용할 수 있다.

이 표들은 표준 L<sup>A</sup>T<sub>E</sub>X 기호 글꼴에서 사용가능한 기호 대부분을 보여준다. 관계 연산자에 대한 부정은 `\not`로 이루어진다.

$G \neq H$

$G \not\equiv H$

### 13.6 $\mathcal{A}\mathcal{M}\mathcal{S}$ -Math

$\mathcal{A}\mathcal{M}\mathcal{S}$ -T<sub>E</sub>X이란 AMS(미국수학회)에서 채택한 T<sub>E</sub>X 매크로 패키지였다. 이것은 L<sup>A</sup>T<sub>E</sub>X과는 독립적으로 발전해오다가, 현재는 L<sup>A</sup>T<sub>E</sub>X에 통합되었다. L<sup>A</sup>T<sub>E</sub>X의 기능이 확장된 대표적인 경우 중의 하나이다.(패키지에 대해서는 16절 참조.) 이것을  $\mathcal{A}\mathcal{M}\mathcal{S}$ -L<sup>A</sup>T<sub>E</sub>X이라고 한다. 특히 `amsmath` 패키지는 종래  $\mathcal{A}\mathcal{M}\mathcal{S}$ -T<sub>E</sub>X의 수식 조판을 L<sup>A</sup>T<sub>E</sub>X 문서에서 활용하게 해준다. 2015년 현재 이것이 L<sup>A</sup>T<sub>E</sub>X

자체 수식 기능을 거의 대체하였다. 그러므로 수식 조판이 문제가 된다면  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Math를 사용하는 것이 좋다. `amsmath`, `amssymb`, `amsthm` 패키지의 사용법을 잘 살펴보고 자신의 문서에 적용하도록 하자. 수학적 조판에 관한 수많은 문제는 이 패키지의 도움을 받으면 대부분 해결된다.

다음은  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Math에서 제공하는 `pmatrix` 환경을 써서 행렬을 조판한 예이다.

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

```
\[
A = \begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
\]
```

## 14 편지

$\text{\LaTeX}$ 으로 간단하게 편지를 작성할 수 있다. 단지 `letter` 형식의 도큐먼트 클래스를 사용하기만 하면 된다. 하나의 입력 파일로 여러 장의 편지를 작성할 수 있다. 본인의 이름과 주소처럼 모든 편지에서 동일하게 사용하는 내용은 파일의 처음에 한 번만 정의해준다. 각각의 편지는 수신인의 주소와 이름을 인자로 하는 `letter` 환경으로 작성된다. 편지의 본문은 `\opening` 명령에 따라 인사와 함께 시작된다.

편지는 `\closing` 명령으로 끝난다. 첨부물이나 사본의 배부처를 표시하기 위해서는 `\enc1`이나 `\cc` 명령어를 사용한다. 명령어 `\closing` 뒤의 모든 내용은 `\ps`를 표기한 후에 기록하여야 한다. 이 명령어는 아무런 표기도 하지 않으므로 사용자는 “추신”이라고 직접 표시하여야 하지만, 추가되는 문장을 정확하게 배치하기 위해서 필요한 명령어이다.

다음의 예제를 살펴보면 좀더 명확하게 이해할 수 있을 것이다.

```
\documentstyle{letter}
\begin{document}

\address{1234 Avenue of the Armadillos \\
        Gnu York, G.Y. 56789}
\signature{R. (Ma) Dillo \\ Director of Cuisine}

\begin{letter}{G. Nathaniel Picking \\
        Acme Exterminators \\
        Illinois}

\opening{Dear Nat,}

I'm afraid that the armadillo problem is still with us.
I did everything ...

... and I hope that you can get rid of the nasty
beasts this time.

\closing{Best Regards,}
\cc{Jimmy Carter\\Richard M. Nixon}
\end{letter}

\end{document}
```

## 15 에러

새로운 L<sup>A</sup>T<sub>E</sub>X 입력 파일을 작성하다보면 아마 실수를 하게 될 것이다. 누구나가 그런 실수는 하는 것이므로 그렇게 염려할 필요는 없다. 실수는 다른 컴퓨터 프로그램에서도 마찬가지이듯이 두 종류가 있다. L<sup>A</sup>T<sub>E</sub>X에서 경고를 해주는 것과 그렇지 않은 것 두 가지 종류이다. 간단히 예를 들자면, 사용자가 본문 입력에서 오타를 냈다면 L<sup>A</sup>T<sub>E</sub>X은 이를 실수라고 알지 못하고 사용자는 결과물을 읽어서 확인한다. 반면에 L<sup>A</sup>T<sub>E</sub>X의 환경명을 잘못 입력했다면 L<sup>A</sup>T<sub>E</sub>X은 어떻게 해야 할지 알 수 없게 된다.

이러한 상황이 발생하면 L<sup>A</sup>T<sub>E</sub>X은 화면에 에러 메시지를 출력한 뒤 처리를 중단하고 사용자의 조치를 기다린다. 에러 메시지가 그다지 친절하지 않기는 해도 어찌되었건 대충 어디에서 무엇이 잘못되었는지를 알려주므로 어디를 살펴보아야 할지 알 수 있다.

`\begin{itemize}`를 `\begin{itemie}`로 잘못 입력하면 어떻게 되는지 살펴보도록 하자. 이러한 지시사항을 만나면 L<sup>A</sup>T<sub>E</sub>X은 다음의 내용을 화면에 출력한다.

```
LaTeX error. See LaTeX manual for explanation.
      Type H <return> for immediate help.
! Environment itemie undefined.
\@latexerr ...for immediate help.}\errmessage {#1}
                                         \endgroup

1.140 \begin{itemie}

?
```

L<sup>A</sup>T<sub>E</sub>X은 ‘?’를 출력한 뒤 사용자가 어떤 지시를 하는지 기다리게 된다.

처음의 두 줄은 단지 L<sup>A</sup>T<sub>E</sub>X이 에러를 감지하였음을 알려주는 것뿐이다. ‘!’로 시작하는 세번째 줄은 *에러 지시문*이다. L<sup>A</sup>T<sub>E</sub>X에 경험이 쌓이기까지는 그 의미가 충분하지는 않겠지만, 이는 어떤 문제가 일어났는지를 알려준다. 이 경우에는 단순히 *itemie*라는 알 수 없는 환경이 나타났음을 말하고 있는 것이다. 다음의 두 줄은 에러가 발생하였을 때 L<sup>A</sup>T<sub>E</sub>X이 무엇을 하고 있었는지를 알려준다. 지금은 별 관계가 없는 내용이므로 무시한다. 마지막 줄은 *에러 위치*를 나타내며, 사용자의 입력 파일에서 문제가 된 줄의 내용을 조금 보여준다. 사용자가 손쉽게 찾을 수 있도록 행번호로 시작한다. 에러가 라인 중간에서 발생하였다면, L<sup>A</sup>T<sub>E</sub>X이 에러라고 인식한 그 부분에서 행을 분리하여 표시한다. 다른 모든 컴퓨터 프로그램과 마찬가지로 L<sup>A</sup>T<sub>E</sub>X에서도 가끔씩 에러가 실제 발생한 위치와 떨어져서 에러 메시지를 보여주기도 하지만 대부분 그리 멀리 있지는 않다.

이 경우 몇 가지 행동을 취할 수 있다. L<sup>A</sup>T<sub>E</sub>X을 잘 알고 있다면 문제를 수정할 수도 있을 것이고, 그렇지 않다면 ‘x’를 입력하여 L<sup>A</sup>T<sub>E</sub>X의 가동을 중단시킨 다음 에러를 수정할 수도 있을 것이다. 제일 좋은 방법은 그냥 리턴키를 누르는 것이다. 그러면 L<sup>A</sup>T<sub>E</sub>X은 마치 아무 일도 없었던 것처럼 계속해서 작동하게 된다. 하나의 실수를 저질렀다면 다른 실수도 저질렀을 수도 있으므로 모든 에러를 한번에 찾고자 할 것이다. 그 편이 에러가 있을 때마다 매번 L<sup>A</sup>T<sub>E</sub>X을 실행해서 하나씩 에러를 수정하는 것보다 훨씬 효율적일 것이다. 나타나는 모든 메시지는 나중의 참고를 위해 *log* 파일에 저장되므로 어떤 에러가 있었는지 기억할 필요는 없다. 일반적으로 문서 자체와 파일 이름이 같고 확장자가 *.log*인 파일이 *log* 파일로 생성되도록 설정되어 있다.

에러가 일어난 행을 살펴보면, 문제가 무엇인지 대략 알 수가 있을 것이다. 무엇이 문제인지 알 수 없을 때에는 아래의 힌트를 살펴보고, 그래도 도움이 되지 않는다면 흔히 발생하는 오류

목록 등을 참고하고 그래도 해결이 되지 않으면 L<sup>A</sup>T<sub>E</sub>X 포럼 등에 질문할 수 있다.

에러를 일으키는 가장 보편적인 실수는 다음과 같다.

- 명령어나 환경의 이름을 잘못 입력하는 경우.
- ‘{’와 ‘}’의 짝이 맞지 않을 경우—항상 짝을 맞추어 사용해야 한다.
- 특수기호 # \$ % & \_ { } ~ ^ \ 를 일반 문자로 그냥 사용한 경우.
- \end 명령어를 빠뜨린 경우.
- 명령어의 인자(중괄호 사이에 있다)를 빠뜨린 경우.

하나의 에러는 L<sup>A</sup>T<sub>E</sub>X을 혼란스럽게 하여 일련의 복잡한 에러를 일으킨다. 이해할 수 있는 에러가 있고 이해할 수 없는 일련의 에러가 연속해 있으면 일단 처음의 에러를 수정하라. 그러면 나머지는 마치 마술처럼 사라질 것이다.

가끔씩 L<sup>A</sup>T<sub>E</sub>X은 에러 메시지도 없이 \*를 출력하고는 정지해버리기도 한다. 다른 것이 원인일 수도 있지만 대부분의 경우에 이것은 \end{document} 명령을 빠뜨려서 일어나는 것이다. 이런 일이 일어나면 \stop이라고 입력하고 리턴키를 눌러라.

마지막으로 L<sup>A</sup>T<sub>E</sub>X은 가끔씩 *warning* 메시지를 출력한다. 이것은 L<sup>A</sup>T<sub>E</sub>X의 작업을 중단할 것까지는 없으나 조사해볼 필요가 있는 문제가 있음을 보고하는 것이다. 대부분이 텍스트 라인의 ‘overfull’과 ‘underfull’ 문제이다. 다음의 메시지는

```
Overfull \hbox (10.58649pt too wide) in paragraph at lines 172--175
[]\tenrm Mathematical for-mu-las may be dis-played. A dis-played
```

단락을 조판함에 있어서 L<sup>A</sup>T<sub>E</sub>X이 행을 분리할 적절한 위치를 찾지 못했음을 알린다. 그 결과 해당 행은 우측 경계 밖으로 빠져 나가게 배치해 버린다. 이 경우에는 10.6포인트만큼이다. 1포인트는 1/72.27 인치이므로, 이를 알아보는 어렵다. 하지만 아무것도 아닌 것은 아니다.

이는 L<sup>A</sup>T<sub>E</sub>X이 줄바꿈에 엄격하기 때문에 표준에 맞지 않는 단락을 생성하느니보다는 맞지 않는 행을 만들어 버리는 것이다. 이 문제에 대한 가장 간단한 해결책은 문제가 된 단락 전체를 \begin{sloppypar}와 \end{sloppypar} 명령어 사이에 넣어버리는 것이다. 이는 본문의 특정 부분에 대해서는 그렇게 엄격하지 않아도 좋다고 L<sup>A</sup>T<sub>E</sub>X에게 알려주는 것이다.

이와는 달리 “Underfull \hboxes” 메시지가 나타날 수도 있다. 이는 단어와 단어 사이에 L<sup>A</sup>T<sub>E</sub>X이 원하는 것보다도 더 많은 공백이 있는 행임을 의미한다. 일반적으로 이 경우에 할 수 있는 일이란 별로 없다. 행이 늘어선 느낌이 있겠지만, 출력물은 괜찮아 보일 것이다. 이와 같은 경우에 할 수 있는 유일한 것은 문제의 단락을 재작성하는 것이다!

## 16 패키지

이 문서의 원본이 쓰여진 1988년의 L<sup>A</sup>T<sub>E</sub>X은 현재의 버전과 달리 \usepackage라는 명령이 없었다. 그래서 이 문서는 패키지에 대해서 별다른 설명을 하고 있지 않다. 그러나 오늘날 L<sup>A</sup>T<sub>E</sub>X의 가장 중요한 핵심 기능이 \usepackage를 이용한 기능의 확장으로 되어 있다. 수많은 패키지들이 제작되었고 텍 설치 프로그램과 함께 사용자의 시스템에 설치된다. 사용자 자신이 스스로 패키지를

제작할 수도 있다. 예를 들면 이 안내서에서는 그림을 문서에 직접 삽입하는 방법을 설명하지 않았는데 그것은 `graphicx`라는 패키지가 제공하는 `\includegraphics`라는 명령을 통해서 하는 것이다. 더 복잡한 수학적 식을 입력하기 위해서 `amsmath`를 써야 하는 경우도 많을 것이다.

패키지는 보통 `.sty`라는 확장자를 갖는다. 예를 들어 `graphicx.sty`를 자신의 문서에서 활용하려면 `\documentclass` 명령과 `\begin{document}` 사이(즉 *preamble*)에 다음과 같은 문장을 둔다.

```
\usepackage{graphicx}
```

패키지 중에서 다음과 같은 것이 많이 사용된다.

`graphicx` 문서에 직접 `.pdf`, `.eps`, `.jpg`, `.png` 그림을 삽입할 수 있게 한다. 그밖에 그림과 그림상자에 관련된 많은 기능을 제공한다.

`kotex` 한글로 작성하는 한국어 문서의 처리를 가능하게 한다.

`amsmath`  $\mathcal{A}\mathcal{M}\mathcal{S}$ -T<sub>E</sub>X 스타일의 수식 조판을 지원한다.

`amsthm` 정리 (theorem), 증명 등과 그 비슷한 유형의 수학 문서 문단 조판을 지원한다.

`color` L<sup>A</sup>T<sub>E</sub>X 문서의 색상에 관련된 문제를 지원한다.

`geometry` L<sup>A</sup>T<sub>E</sub>X 문서의 판형과 판면, 용지크기와 여백 등에 관한 기능을 제공한다.

`makeidx` `\index` 명령을 통하여 색인 (index)을 작성할 수 있게 한다.

`setspace` 행간격, 문단간격 등 간격에 관련된 제어 매크로를 제공한다.

오늘날 L<sup>A</sup>T<sub>E</sub>X 문서 작성은 패키지의 지원을 받아 더욱 자유롭고 풍성하게 되었다. 그러므로 자신에게 필요한 패키지가 어떤 것인지 잘 알아보는 것이 좋다.

이 안내서에서 설명한 것을 넘어서서 L<sup>A</sup>T<sub>E</sub>X의 기능을 확장하는 무수한 패키지들이 있다.

패키지들은 각각의 사용법과 자체 정의 명령들이 있다. 패키지를 사용할 때는 반드시 그 패키지 설명 문서를 `texdoc` 명령으로 읽어보기 바란다. 한국어 문서 작성을 위해 `kotexdoc`과 `xetexko-doc`을 읽어야 하는 것도 마찬가지다.

## 17 마치는 글

이제 다양한 종류의 문서를 작성할 수 있을 정도로 L<sup>A</sup>T<sub>E</sub>X에 대해 많이 알게 되었을 것이다. 하지만 이 문서에서 다룬 것은 L<sup>A</sup>T<sub>E</sub>X이 할 수 있는 일에 비한다면 겨우 수박의 겉을 훑은 것에 지나지 않는다. 이 문서 전체를 L<sup>A</sup>T<sub>E</sub>X으로 작성하였다(속임수나 복사기를 사용하지 않고서). 게다가 사용할 수 있는 모든 특성을 다 사용한 것도 아니다. 이로 미루어 여러분의 처분에 맡겨져 있는 L<sup>A</sup>T<sub>E</sub>X의 파워를 조금이나마 느낄 수 있을 것이다.

L<sup>A</sup>T<sub>E</sub>X으로 문서를 작성할 때는 항상 도움말을 옆에 두는 것이 좋다. 일반적인 개설서, *The L<sup>A</sup>T<sub>E</sub>X Companion* [3], 각 패키지 안내서 등을 꾸준히 참조해야 한다.



---

일반적인 문서 작성에는 이 정도로 충분하다. 그러나  $\LaTeX$ 은 결국  $\TeX$ 으로 구축된 매크로 패키지이므로,  $\TeX$  자체에 대해서 깊이있는 이해를 하여 전문가가 되려 한다면 Knuth 교수의 *TeXbook* [2]을 읽어보아야 할 것이다.

그리고 또다른 경고 한 마디,  $\LaTeX$ 을 가지고 놀다보면 여러분의 문서가 결코 다시 예전과 같아질 수 없다…….

## 참고 문헌

- [1] Leslie Lamport, *LaTeX—A Document Preparation System—User’s Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.
- [2] Donald Knuth, *The TeXbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986.
- [3] Frank Mittelbach and Michel Goossens, *The LaTeX Companion*, 2nd edition, Addison-Wesley, MA, USA, 2005.

## 부록 A 수학 기호

이 부록은 lshort에 있는 것을 가져다 썼다. 원래 David Carlisle 씨가 작성한 `symbols.tex`을 토대로 해서 Josef Tkadlec 씨의 확장 제안을 받아들여 만들어진 표라고 한다.

완전한 기호문자의 목록을 보려면 *The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbols*라는 문서를 읽어보기 바란다. `texdoc symbols-a4` 명령으로 데스크탑에서 바로 읽을 수 있다.

표 A.1: 수학 모드의 액센트 기호.

$\hat{a}$	<code>\hat{a}</code>	$\check{a}$	<code>\check{a}</code>	$\tilde{a}$	<code>\tilde{a}</code>	$\acute{a}$	<code>\acute{a}</code>
$\grave{a}$	<code>\grave{a}</code>	$\dot{a}$	<code>\dot{a}</code>	$\ddot{a}$	<code>\ddot{a}</code>	$\breve{a}$	<code>\breve{a}</code>
$\bar{a}$	<code>\bar{a}</code>	$\vec{a}$	<code>\vec{a}</code>	$\widehat{A}$	<code>\widehat{A}</code>	$\widetilde{A}$	<code>\widetilde{A}</code>

표 A.2: 그리스 문자 소문자.

$\alpha$	<code>\alpha</code>	$\theta$	<code>\theta</code>	$o$	<code>o</code>	$\upsilon$	<code>\upsilon</code>
$\beta$	<code>\beta</code>	$\vartheta$	<code>\vartheta</code>	$\pi$	<code>\pi</code>	$\phi$	<code>\phi</code>
$\gamma$	<code>\gamma</code>	$\iota$	<code>\iota</code>	$\varpi$	<code>\varpi</code>	$\varphi$	<code>\varphi</code>
$\delta$	<code>\delta</code>	$\kappa$	<code>\kappa</code>	$\rho$	<code>\rho</code>	$\chi$	<code>\chi</code>
$\epsilon$	<code>\epsilon</code>	$\lambda$	<code>\lambda</code>	$\varrho$	<code>\varrho</code>	$\psi$	<code>\psi</code>
$\varepsilon$	<code>\varepsilon</code>	$\mu$	<code>\mu</code>	$\sigma$	<code>\sigma</code>	$\omega$	<code>\omega</code>
$\zeta$	<code>\zeta</code>	$\nu$	<code>\nu</code>	$\varsigma$	<code>\varsigma</code>		
$\eta$	<code>\eta</code>	$\xi$	<code>\xi</code>	$\tau$	<code>\tau</code>		

표 A.3: 그리스 문자 대문자.

$\Gamma$	<code>\Gamma</code>	$\Lambda$	<code>\Lambda</code>	$\Sigma$	<code>\Sigma</code>	$\Psi$	<code>\Psi</code>
$\Delta$	<code>\Delta</code>	$\Xi$	<code>\Xi</code>	$\Upsilon$	<code>\Upsilon</code>	$\Omega$	<code>\Omega</code>
$\Theta$	<code>\Theta</code>	$\Pi$	<code>\Pi</code>	$\Phi$	<code>\Phi</code>		

표 A.4: Binary Relations.

다음 부호에는 `\not` 명령을 앞에 붙여서 부정 기호로 만들 수 있다.

$<$	<code>&lt;</code>	$>$	<code>&gt;</code>	$=$	<code>=</code>
$\leq$	<code>\leq or \le</code>	$\geq$	<code>\geq or \ge</code>	$\equiv$	<code>\equiv</code>
$\ll$	<code>\ll</code>	$\gg$	<code>\gg</code>	$\doteq$	<code>\doteq</code>
$\prec$	<code>\prec</code>	$\succ$	<code>\succ</code>	$\sim$	<code>\sim</code>
$\preceq$	<code>\preceq</code>	$\succeq$	<code>\succeq</code>	$\simeq$	<code>\simeq</code>
$\subset$	<code>\subset</code>	$\supset$	<code>\supset</code>	$\approx$	<code>\approx</code>
$\subseteq$	<code>\subseteq</code>	$\supseteq$	<code>\supseteq</code>	$\cong$	<code>\cong</code>
$\sqsubset$	<code>\sqsubset <sup>a</sup></code>	$\sqsupset$	<code>\sqsupset <sup>a</sup></code>	$\Join$	<code>\Join <sup>a</sup></code>
$\sqsubseteq$	<code>\sqsubseteq</code>	$\sqsupseteq$	<code>\sqsupseteq</code>	$\bowtie$	<code>\bowtie</code>
$\in$	<code>\in</code>	$\ni$	<code>\ni , \owns</code>	$\propto$	<code>\propto</code>
$\vdash$	<code>\vdash</code>	$\dashv$	<code>\dashv</code>	$\models$	<code>\models</code>
$ $	<code>\mid</code>	$\parallel$	<code>\parallel</code>	$\perp$	<code>\perp</code>
$\smile$	<code>\smile</code>	$\frown$	<code>\frown</code>	$\asymp$	<code>\asymp</code>
$:$	<code>:</code>	$\notin$	<code>\notin</code>	$\neq$	<code>\neq or \ne</code>

<sup>a</sup>Use the `latexsym` package to access this symbol

표 A.5: Binary Operators.

$+$	<code>+</code>	$-$	<code>-</code>		
$\pm$	<code>\pm</code>	$\mp$	<code>\mp</code>	$\triangleleft$	<code>\triangleleft</code>
$\cdot$	<code>\cdot</code>	$\div$	<code>\div</code>	$\triangleright$	<code>\triangleright</code>
$\times$	<code>\times</code>	$\setminus$	<code>\setminus</code>	$\star$	<code>\star</code>
$\cup$	<code>\cup</code>	$\cap$	<code>\cap</code>	$*$	<code>\ast</code>
$\sqcup$	<code>\sqcup</code>	$\sqcap$	<code>\sqcap</code>	$\circ$	<code>\circ</code>
$\vee$	<code>\vee , \lor</code>	$\wedge$	<code>\wedge , \land</code>	$\bullet$	<code>\bullet</code>
$\oplus$	<code>\oplus</code>	$\ominus$	<code>\ominus</code>	$\diamond$	<code>\diamond</code>
$\odot$	<code>\odot</code>	$\oslash$	<code>\oslash</code>	$\uplus$	<code>\uplus</code>
$\otimes$	<code>\otimes</code>	$\bigcirc$	<code>\bigcirc</code>	$\amalg$	<code>\amalg</code>
$\triangleup$	<code>\bigtriangleup</code>	$\triangledown$	<code>\bigtriangledown</code>	$\dagger$	<code>\dagger</code>
$\triangleleft$	<code>\lhd <sup>a</sup></code>	$\triangleright$	<code>\rhd <sup>a</sup></code>	$\ddagger$	<code>\ddagger</code>
$\trianglelefteq$	<code>\unlhd <sup>a</sup></code>	$\trianglerighteq$	<code>\unrhd <sup>a</sup></code>	$\wr$	<code>\wr</code>

표 A.6: BIG Operators.

$\sum$	<code>\sum</code>	$\bigcup$	<code>\bigcup</code>	$\bigvee$	<code>\bigvee</code>	$\bigoplus$	<code>\bigoplus</code>
$\prod$	<code>\prod</code>	$\bigcap$	<code>\bigcap</code>	$\bigwedge$	<code>\bigwedge</code>	$\bigotimes$	<code>\bigotimes</code>
$\coprod$	<code>\coprod</code>	$\bigsqcup$	<code>\bigsqcup</code>			$\bigodot$	<code>\bigodot</code>
$\int$	<code>\int</code>	$\oint$	<code>\oint</code>			$\biguplus$	<code>\biguplus</code>

표 A.7: 화살표.

$\leftarrow$	<code>\leftarrow</code> or <code>\gets</code>	$\longleftarrow$	<code>\longleftarrow</code>	$\uparrow$	<code>\uparrow</code>
$\rightarrow$	<code>\rightarrow</code> or <code>\to</code>	$\longrightarrow$	<code>\longrightarrow</code>	$\downarrow$	<code>\downarrow</code>
$\leftrightarrow$	<code>\leftrightarrow</code>	$\longleftrightarrow$	<code>\longleftrightarrow</code>	$\updownarrow$	<code>\updownarrow</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\Llongleftarrow$	<code>\Llongleftarrow</code>	$\Uparrow$	<code>\Uparrow</code>
$\Rrightarrow$	<code>\Rrightarrow</code>	$\Rlongrightarrow$	<code>\Rlongrightarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\Leftrightarrow$	<code>\Leftrightarrow</code>	$\Llongleftrightarrow$	<code>\Llongleftrightarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\mapsto$	<code>\mapsto</code>	$\longmapsto$	<code>\longmapsto</code>	$\nearrow$	<code>\nearrow</code>
$\hookrightarrow$	<code>\hookrightarrow</code>	$\hookrightarrow$	<code>\hookrightarrow</code>	$\searrow$	<code>\searrow</code>
$\leftharpoonup$	<code>\leftharpoonup</code>	$\rightharpoonup$	<code>\rightharpoonup</code>	$\swarrow$	<code>\swarrow</code>
$\leftharpoondown$	<code>\leftharpoondown</code>	$\rightharpoondown$	<code>\rightharpoondown</code>	$\nwarrow$	<code>\nwarrow</code>
$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\iff$ (bigger spaces)	<code>\iff</code> (bigger spaces)	$\leadsto$	<code>\leadsto</code> <sup>a</sup>

<sup>a</sup>Use the latexsym package to access this symbol

표 A.8: 짝맞춤 문자(Delimiters).

$($	<code>(</code>	$)$	<code>)</code>	$\uparrow$	<code>\uparrow</code>	$\Uparrow$	<code>\Uparrow</code>
$[$	<code>[</code> or <code>\lbrack</code>	$]$	<code>]</code> or <code>\rbrack</code>	$\downarrow$	<code>\downarrow</code>	$\Downarrow$	<code>\Downarrow</code>
$\{$	<code>\{</code> or <code>\lbrace</code>	$\}$	<code>\}</code> or <code>\rbrace</code>	$\updownarrow$	<code>\updownarrow</code>	$\Updownarrow$	<code>\Updownarrow</code>
$\langle$	<code>\langle</code>	$\rangle$	<code>\rangle</code>	$ $	<code> </code> or <code>\vert</code>	$\ $	<code>\ </code> or <code>\Vert</code>
$\lfloor$	<code>\lfloor</code>	$\rfloor$	<code>\rfloor</code>	$\lceil$	<code>\lceil</code>	$\rceil$	<code>\rceil</code>
$/$	<code>/</code>	$\backslash$	<code>\backslash</code>	.	(dual. empty)		

표 A.9: 큰 짝맞춤 문자(Large Delimiters).

$\left($	<code>\lgroup</code>	$\right)$	<code>\rgroup</code>	$\left[$	<code>\lgroup</code>	$\right]$	<code>\rgroup</code>
$\uparrow$	<code>\arrowvert</code>	$\downarrow$	<code>\Arrowvert</code>	$\lceil$	<code>\lgroup</code>	$\rceil$	<code>\rgroup</code>
				$\lfloor$	<code>\lgroup</code>	$\rfloor$	<code>\rgroup</code>

표 A.10: 그밖의 기호.

$\dots$	<code>\dots</code>	$\cdots$	<code>\cdots</code>	$\vdots$	<code>\vdots</code>	$\ddots$	<code>\ddots</code>
$\hbar$	<code>\hbar</code>	$\imath$	<code>\imath</code>	$\jmath$	<code>\jmath</code>	$\ell$	<code>\ell</code>
$\Re$	<code>\Re</code>	$\Im$	<code>\Im</code>	$\aleph$	<code>\aleph</code>	$\wp$	<code>\wp</code>
$\forall$	<code>\forall</code>	$\exists$	<code>\exists</code>	$\mho$	<code>\mho</code> <sup>a</sup>	$\partial$	<code>\partial</code>
$'$	<code>'</code>	$\prime$	<code>\prime</code>	$\emptyset$	<code>\emptyset</code>	$\infty$	<code>\infty</code>
$\nabla$	<code>\nabla</code>	$\triangle$	<code>\triangle</code>	$\square$	<code>\square</code> <sup>a</sup>	$\diamond$	<code>\diamond</code> <sup>a</sup>
$\perp$	<code>\perp</code>	$\top$	<code>\top</code>	$\angle$	<code>\angle</code>	$\surd$	<code>\surd</code>
$\diamondsuit$	<code>\diamondsuit</code>	$\heartsuit$	<code>\heartsuit</code>	$\clubsuit$	<code>\clubsuit</code>	$\spadesuit$	<code>\spadesuit</code>
$\neg$	<code>\neg</code> or <code>\not</code>	$\flat$	<code>\flat</code>	$\natural$	<code>\natural</code>	$\sharp$	<code>\sharp</code>

<sup>a</sup>Use the latexsym package to access this symbol

## 표 A.11: Non-Mathematical Symbols.

다음 기호들은 텍스트 모드에서도 사용할 수 있다.

†	<code>\dag</code>	§	<code>\S</code>	©	<code>\copyright</code>	®	<code>\textregistered</code>
‡	<code>\ddag</code>	¶	<code>\P</code>	£	<code>\pounds</code>	%	<code>\%</code>

## 표 A.12: AMS Delimiters.

⌈	<code>\ulcorner</code>	⌋	<code>\urcorner</code>	⌌	<code>\llcorner</code>	⌍	<code>\lrcorner</code>
	<code>\lvert</code>		<code>\rvert</code>		<code>\lVert</code>		<code>\rVert</code>

## 표 A.13: AMS Greek and Hebrew.

$\digamma$	<code>\digamma</code>	$\varkappa$	<code>\varkappa</code>	$\beth$	<code>\beth</code>	$\gimel$	<code>\gimel</code>	$\daleth$	<code>\daleth</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	----------	---------------------	-----------	----------------------

## 표 A.14: AMS Binary Relations.

$\lessdot$	<code>\lessdot</code>	$\gtrdot$	<code>\gtrdot</code>	$\doteqdot$ or $\Doteq$	<code>\doteqdot</code> or <code>\Doteq</code>
$\leqslant$	<code>\leqslant</code>	$\geqslant$	<code>\geqslant</code>	$\risingdotseq$	<code>\risingdotseq</code>
$\leqslantless$	<code>\leqslantless</code>	$\eqslantgtr$	<code>\eqslantgtr</code>	$\fallingdotseq$	<code>\fallingdotseq</code>
$\leqq$	<code>\leqq</code>	$\geqq$	<code>\geqq</code>	$\eqcirc$	<code>\eqcirc</code>
$\lll$ or $\llless$	<code>\lll</code> or <code>\llless</code>	$\ggg$ or $\gggtr$	<code>\ggg</code> or <code>\gggtr</code>	$\circeq$	<code>\circeq</code>
$\lesssim$	<code>\lesssim</code>	$\gtrsim$	<code>\gtrsim</code>	$\triangleq$	<code>\triangleq</code>
$\lessapprox$	<code>\lessapprox</code>	$\gtrapprox$	<code>\gtrapprox</code>	$\bumpeq$	<code>\bumpeq</code>
$\lessgtr$	<code>\lessgtr</code>	$\gtrless$	<code>\gtrless</code>	$\Bumpeq$	<code>\Bumpeq</code>
$\lesseqgtr$	<code>\lesseqgtr</code>	$\gtreqless$	<code>\gtreqless</code>	$\thicksim$	<code>\thicksim</code>
$\lesseqqgtr$	<code>\lesseqqgtr</code>	$\gtreqqlless$	<code>\gtreqqlless</code>	$\thickapprox$	<code>\thickapprox</code>
$\preccurlyeq$	<code>\preccurlyeq</code>	$\succcurlyeq$	<code>\succcurlyeq</code>	$\approxeq$	<code>\approxeq</code>
$\curlyeqprec$	<code>\curlyeqprec</code>	$\curlyeqsucc$	<code>\curlyeqsucc</code>	$\backsim$	<code>\backsim</code>
$\precsim$	<code>\precsim</code>	$\succsim$	<code>\succsim</code>	$\backsimeq$	<code>\backsimeq</code>
$\precapprox$	<code>\precapprox</code>	$\succapprox$	<code>\succapprox</code>	$\vDash$	<code>\vDash</code>
$\subseteqq$	<code>\subseteqq</code>	$\supseteqq$	<code>\supseteqq</code>	$\Vdash$	<code>\Vdash</code>
$\Subset$	<code>\Subset</code>	$\Supset$	<code>\Supset</code>	$\Vvdash$	<code>\Vvdash</code>
$\sqsubset$	<code>\sqsubset</code>	$\sqsupset$	<code>\sqsupset</code>	$\backepsilon$	<code>\backepsilon</code>
$\therefore$	<code>\therefore</code>	$\because$	<code>\because</code>	$\varpropto$	<code>\varpropto</code>
$\shortmid$	<code>\shortmid</code>	$\shortparallel$	<code>\shortparallel</code>	$\between$	<code>\between</code>
$\smallsmile$	<code>\smallsmile</code>	$\smallfrown$	<code>\smallfrown</code>	$\pitchfork$	<code>\pitchfork</code>
$\vartriangleleft$	<code>\vartriangleleft</code>	$\vartriangleright$	<code>\vartriangleright</code>	$\blacktriangleleft$	<code>\blacktriangleleft</code>
$\trianglelefteq$	<code>\trianglelefteq</code>	$\trianglerighteq$	<code>\trianglerighteq</code>	$\blacktriangleright$	<code>\blacktriangleright</code>

## ⌘ A.15: AMS Arrows.

$\dashleftarrow$	<code>\dashleftarrow</code>	$\dashrightarrow$	<code>\dashrightarrow</code>	$\multimap$	<code>\multimap</code>
$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>	$\rightrightarrows$	<code>\rightrightarrows</code>	$\Uparrow$	<code>\upuparrows</code>
$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>	$\rightleftarrows$	<code>\rightleftarrows</code>	$\Downarrow$	<code>\downdownarrows</code>
$\Lleftarrow$	<code>\Lleftarrow</code>	$\Rrightarrow$	<code>\Rrightarrow</code>	$\Uparrow$	<code>\upharpoonleft</code>
$\twoheadleftarrow$	<code>\twoheadleftarrow</code>	$\twoheadrightarrow$	<code>\twoheadrightarrow</code>	$\Uparrow$	<code>\upharpoonright</code>
$\leftarrowtail$	<code>\leftarrowtail</code>	$\rightarrowtail$	<code>\rightarrowtail</code>	$\Downarrow$	<code>\downharpoonleft</code>
$\leftrightharpoons$	<code>\leftrightharpoons</code>	$\rightleftharpoons$	<code>\rightleftharpoons</code>	$\Downarrow$	<code>\downharpoonright</code>
$\Lsh$	<code>\Lsh</code>	$\Rsh$	<code>\Rsh</code>	$\rightsquigarrow$	<code>\rightsquigarrow</code>
$\looparrowleft$	<code>\looparrowleft</code>	$\looparrowright$	<code>\looparrowright</code>	$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>
$\curvearrowleft$	<code>\curvearrowleft</code>	$\curvearrowright$	<code>\curvearrowright</code>		
$\circlearrowleft$	<code>\circlearrowleft</code>	$\circlearrowright$	<code>\circlearrowright</code>		

## ⌘ A.16: AMS Negated Binary Relations and Arrows.

$\nless$	<code>\nless</code>	$\ngtr$	<code>\ngtr</code>	$\varsubsetneqq$	<code>\varsubsetneqq</code>
$\lneq$	<code>\lneq</code>	$\gneq$	<code>\gneq</code>	$\varsupsetneqq$	<code>\varsupsetneqq</code>
$\nleq$	<code>\nleq</code>	$\ngeq$	<code>\ngeq</code>	$\subsetneqq$	<code>\subsetneqq</code>
$\nleqslant$	<code>\nleqslant</code>	$\ngeqslant$	<code>\ngeqslant</code>	$\supsetneqq$	<code>\supsetneqq</code>
$\lneqq$	<code>\lneqq</code>	$\gneqq$	<code>\gneqq</code>	$\mid$	<code>\mid</code>
$\lvertneqq$	<code>\lvertneqq</code>	$\gvertneqq$	<code>\gvertneqq</code>	$\parallel$	<code>\parallel</code>
$\nleqq$	<code>\nleqq</code>	$\ngeqq$	<code>\ngeqq</code>	$\shortmid$	<code>\shortmid</code>
$\lnsim$	<code>\lnsim</code>	$\gnsim$	<code>\gnsim</code>	$\shortparallel$	<code>\shortparallel</code>
$\lnapprox$	<code>\lnapprox</code>	$\gnapprox$	<code>\gnapprox</code>	$\sim$	<code>\sim</code>
$\nprec$	<code>\nprec</code>	$\nsucc$	<code>\nsucc</code>	$\cong$	<code>\cong</code>
$\npreceq$	<code>\npreceq</code>	$\nsucceq$	<code>\nsucceq</code>	$\nvdash$	<code>\nvdash</code>
$\precneqq$	<code>\precneqq</code>	$\succneqq$	<code>\succneqq</code>	$\nvDash$	<code>\nvDash</code>
$\precnsim$	<code>\precnsim</code>	$\succnsim$	<code>\succnsim</code>	$\nVdash$	<code>\nVdash</code>
$\precnapprox$	<code>\precnapprox</code>	$\succnapprox$	<code>\succnapprox</code>	$\nVDash$	<code>\nVDash</code>
$\subsetneq$	<code>\subsetneq</code>	$\supsetneq$	<code>\supsetneq</code>	$\triangleleft$	<code>\triangleleft</code>
$\varsubsetneq$	<code>\varsubsetneq</code>	$\varsupsetneq$	<code>\varsupsetneq</code>	$\triangleright$	<code>\triangleright</code>
$\subsetseteq$	<code>\subsetseteq</code>	$\nsupseteq$	<code>\nsupseteq</code>	$\trianglelefteq$	<code>\trianglelefteq</code>
$\subsetneqq$	<code>\subsetneqq</code>	$\supsetneqq$	<code>\supsetneqq</code>	$\trianglerighteq$	<code>\trianglerighteq</code>
$\nleftarrow$	<code>\nleftarrow</code>	$\nrightarrow$	<code>\nrightarrow</code>	$\leftrightsquigarrow$	<code>\leftrightsquigarrow</code>
$\nLeftarrow$	<code>\nLeftarrow</code>	$\nRightarrow$	<code>\nRightarrow</code>	$\Leftrightarrow$	<code>\Leftrightarrow</code>

## ⌘ A.17: AMS Binary Operators.

$\dot{+}$	<code>\dotplus</code>	$\cdot$	<code>\centerdot</code>	$\intercal$	<code>\intercal</code>
$\ltimes$	<code>\ltimes</code>	$\rtimes$	<code>\rtimes</code>	$\div$	<code>\divideontimes</code>
$\Cup$	<code>\Cup</code> or <code>\doublecup</code>	$\Cap$	<code>\Cap</code> or <code>\doublecap</code>	$\smallsetminus$	<code>\smallsetminus</code>
$\veebar$	<code>\veebar</code>	$\bar{\wedge}$	<code>\barwedge</code>	$\overline{\wedge}$	<code>\doublebarwedge</code>
$\boxplus$	<code>\boxplus</code>	$\boxminus$	<code>\boxminus</code>	$\ominus$	<code>\circleddash</code>
$\boxtimes$	<code>\boxtimes</code>	$\boxdot$	<code>\boxdot</code>	$\odot$	<code>\circledcirc</code>
$\leftthreetimes$	<code>\leftthreetimes</code>	$\rightthreetimes$	<code>\rightthreetimes</code>	$\circledast$	<code>\circledast</code>
$\curlyvee$	<code>\curlyvee</code>	$\curlywedge$	<code>\curlywedge</code>		

## ⌘ A.18: AMS Miscellaneous.

$\hbar$	<code>\hbar</code>	$\hslash$	<code>\hslash</code>	$\Bbbk$	<code>\Bbbk</code>
$\square$	<code>\square</code>	$\blacksquare$	<code>\blacksquare</code>	$\textcircled{S}$	<code>\circledS</code>
$\vartriangle$	<code>\vartriangle</code>	$\blacktriangle$	<code>\blacktriangle</code>	$\complement$	<code>\complement</code>
$\nabla$	<code>\triangledown</code>	$\blacktriangledown$	<code>\blacktriangledown</code>	$\Game$	<code>\Game</code>
$\lozenge$	<code>\lozenge</code>	$\blacklozenge$	<code>\blacklozenge</code>	$\bigstar$	<code>\bigstar</code>
$\angle$	<code>\angle</code>	$\measuredangle$	<code>\measuredangle</code>	$\sphericalangle$	<code>\sphericalangle</code>
$\diagup$	<code>\diagup</code>	$\diagdown$	<code>\diagdown</code>	$\backprime$	<code>\backprime</code>
$\nexists$	<code>\nexists</code>	$\Finv$	<code>\Finv</code>	$\varnothing$	<code>\varnothing</code>
$\eth$	<code>\eth</code>	$\mho$	<code>\mho</code>		

## ⌘ A.19: Math Alphabets.

Example	Command	Required package
$\mathrm{ABCdef}$	<code>\mathrm{ABCdef}</code>	
$\mathit{ABCdef}$	<code>\mathit{ABCdef}</code>	
$\mathnormal{ABCdef}$	<code>\mathnormal{ABCdef}</code>	
$\mathcal{ABC}$	<code>\mathcal{ABC}</code>	euscript with option <code>mathcal</code>
$\mathfrak{ABCdef}$	<code>\mathfrak{ABCdef}</code>	eufrak
$\mathbb{ABC}$	<code>\mathbb{ABC}</code>	amsfonts or amssymb

## 부록 B 학습을 위한 엄청난 수학 예제

$$\phi(t) = \frac{1}{\sqrt{2\pi}} \int_0^t e^{-x^2/2} dx \quad (\text{B.1})$$

```
\begin{equation}
\phi(t)=\frac{1}{\sqrt{2\pi}}
\int^t_0 e^{-x^2/2} dx
\end{equation}
```

$$\prod_{j \geq 0} \left( \sum_{k \geq 0} a_{jk} z^k \right) = \sum_{k \geq 0} z^n \left( \sum_{\substack{k_0, k_1, \dots \geq 0 \\ k_0 + k_1 + \dots = n}} a_{0k_0} a_{1k_1} \dots \right) \quad (\text{B.2})$$

```
\begin{equation}
\prod_{j \geq 0}
\left( \sum_{k \geq 0} a_{jk} z^k \right)
= \sum_{k \geq 0} z^n
\left( \sum_{\substack{k_0, k_1, \dots \geq 0 \\ k_0 + k_1 + \dots = n}}
a_{0k_0} a_{1k_1} \dots \right)
\end{equation}
```

$$\pi(n) = \sum_{m=2}^n \left[ \left( \sum_{k=1}^{m-1} \lfloor (m/k) / \lceil m/k \rceil \right)^{-1} \right] \quad (\text{B.3})$$

```
\begin{equation}
\pi(n) = \sum_{m=2}^n
\left[ \left( \sum_{k=1}^{m-1}
\lfloor (m/k) / \lceil m/k \rceil
\rfloor \right)^{-1} \right]
\end{equation}
```

$$\underbrace{\{a_1, \dots, a_k\}}_{k \text{ a's}} \underbrace{\{b_1, \dots, b_l\}}_{l \text{ b's}} \quad (\text{B.4})$$

$k+1$  elements

```
\begin{equation}
\{\underbrace{\%
\overbrace{\mathstrut a, \dots, a}\%
~{k} \text{ a's},
\overbrace{\mathstrut b, \dots, b}\%
~{l} \text{ b's}}\}_{k+1} \mathrm{elements} \}
\end{equation}
```

$$W^+ \begin{array}{l} \nearrow \mu^+ + \nu_\mu \\ \rightarrow \pi^+ + \pi^0 \\ \rightarrow \kappa^+ + \pi^0 \\ \searrow e^+ + \nu_e \end{array}$$

```
\begin{displaymath}
\mbox{W}^+ \! \! \! \left[ \begin{array}{l}
\nearrow \mu^+ + \nu_\mu \\
\rightarrow \pi^+ + \pi^0 \\
\rightarrow \kappa^+ + \pi^0 \\
\searrow e^+ + \nu_e \end{array} \right]
\end{displaymath}
```



$$F(x, y) = 0 \quad \text{and} \quad \begin{vmatrix} F''_{xx} & F''_{xy} & F'_x \\ F''_{yx} & F''_{yy} & F'_y \\ F'_x & F'_y & 0 \end{vmatrix} = 0$$

```
\begin{displaymath}
{F}(x,y)=0\quad\mathrm{and}\quad\quad
\left|\begin{array}{ccc}
F_{xx}'' & F_{xy}'' & F_x' \\
F_{yx}'' & F_{yy}'' & F_y' \\
F_x' & F_y' & 0
\end{array}\right|=0
\end{displaymath}
```

$$\pm \frac{\begin{vmatrix} x_1 - x_2 & y_1 - y_2 & z_1 - z_2 \\ l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \end{vmatrix}}{\sqrt{\begin{vmatrix} l_1 & m_1 \\ l_2 & m_2 \end{vmatrix}^2 + \begin{vmatrix} m_1 & n_1 \\ n_1 & l_1 \end{vmatrix}^2 + \begin{vmatrix} m_2 & n_2 \\ n_2 & l_2 \end{vmatrix}^2}}$$

```
\begin{displaymath}
\frac{\pm
\left|\begin{array}{ccc}
x_1-x_2 & y_1-y_2 & z_1-z_2 \\
l_1 & m_1 & n_1 \\
l_2 & m_2 & n_2
\end{array}\right|}{
\sqrt{\left|\begin{array}{cc}
l_1&m_1 \\
l_2&m_2
\end{array}\right|^2
+\left|\begin{array}{cc}
m_1&n_1 \\
n_1&l_1
\end{array}\right|^2
+\left|\begin{array}{cc}
m_2&n_2 \\
n_2&l_2
\end{array}\right|^2}}
\end{displaymath}
```

$$\sigma_0^f(Q, T_{3R}, \beta, s) = \frac{4\pi\alpha^2}{3s} \beta \times \left[ \frac{Q^2 \left\{ \frac{3-\beta^2}{2} \right\} - 2QC_V C'_V s (s - M_Z^2)}{(s - M_Z^2)^2 + M_Z^2 \Gamma_Z^2 \left\{ \frac{3-\beta^2}{2} \right\}} + \frac{(C_V^2 + C_A^2) s^2}{(s - M_Z^2)^2 + M_Z^2 \Gamma_Z^2 \left\{ C_V'^2 \left\{ \frac{3-\beta^2}{2} \right\} + C_A'^2 \{\beta^2\} \right\}} \right] \quad (\text{B.5})$$

```
\newcommand{\CA}{C_{\rm A}} \newcommand{\CV}{C_{\rm V}}
\newcommand{\CPA}{C'_{\rm A}} \newcommand{\CPV}{C'_{\rm V}}
\newcommand{\GZ}{\Gamma^2_{\rm Z}}
\newcommand{\MZ}{M^2_{\rm Z}} \newcommand{\MZs}{(s-M^2_{\rm Z})}
\newcommand{\BE}{\left\{\frac{\displaystyle 3-\beta^2}{\displaystyle 2}\right\}}
\begin{eqnarray}
\sigma^f_0(Q,T_{3R},\beta,s) & = &
\frac{4\pi\alpha^2}{3s}\beta \times
\left[ \frac{Q^2 \ \BE - 2Q \ \CV \ \CPV \ s \ \MZs}{\right.}
& & \left. \frac{(C_V^2 + C_A^2) s^2}{(s - M_Z^2)^2 + M_Z^2 \Gamma_Z^2 \left\{ C_V'^2 \left\{ \frac{3-\beta^2}{2} \right\} + C_A'^2 \{\beta^2\} \right\}} \right]
\end{eqnarray}
```

## 부록 C 역자의 말

이 문서에 대하여. 이 문서는 J. Warbrick 씨가 작성한 문서를 토대로 David Carlisle 씨 등이 수식에 관한 내용 등을 보충하여 만든 “Essential L<sup>A</sup>T<sub>E</sub>X++”을 번역한 것이다.

이 문서의 번역은 김재우 님이 한 것으로, 1995–6년 사이에 번역이 이루어졌고, 처음에는 도은이네 집에서 배포되었다. 그 후, KTUG의 문서 서비스의 일환으로 제공되던 것을 김강수가 새롭게 .pdf 파일로 구성하였다.

2005년에 dhucs를 사용하여 한글 문서로 구성된 이래, 10년을 기다려 다시 개정하였다. 이번에는 X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X을 사용하였고 한글 폰트는 함초롬 LVT를 적용하였다.

이 문서는 L<sup>A</sup>T<sub>E</sub>X의 기초 중의 기초인 `small2e.tex`의 설명서이다. 맨처음 L<sup>A</sup>T<sub>E</sub>X에 입문할 때 반드시 읽어야 할 문서라고 생각한다.

이 글을 읽은 후에 L<sup>A</sup>T<sub>E</sub>X의 작용 방식에 관해서 어느 정도 이해하였다면, 이보다 조금 더 자세한 참고문서를 읽어볼 것을 권장한다.