

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

Jon Warbrick

with additions by David Carlisle, Michel Goossens, Sebastian Rahtz, Adrian Clark

김 재 우 옮김

January 1994

## 목 차

1	개요 . . . . .	4
2	L <sup>A</sup> T <sub>E</sub> X은 어떻게 움직이는가? . . . . .	4
1	Simple Text . . . . .	6
3	샘플 L <sup>A</sup> T <sub>E</sub> X 화일 . . . . .	6
4	도큐먼트 클래스와 옵션 . . . . .	8
5	환경 . . . . .	8
6	글꼴 . . . . .	10
7	섹션 명령과 목차 . . . . .	11
8	특수기호의 식자 . . . . .	12
9	제목 . . . . .	12
10	표의 사용 . . . . .	13
11	표와 그림 . . . . .	14
12	상호참조와 인용 . . . . .	15
13	수식의 조판 . . . . .	16
14	\$가 의미 하는 것은? . . . . .	18
15	편지 . . . . .	19
16	에러 . . . . .	20
17	마치는 글 . . . . .	21
A	수학기호 . . . . .	22
B	학습을 위한 엄청난 수학 예제 . . . . .	27
	Bibliography . . . . .	29

## 표 목차

1	Greek letters . . . . .	22
2	Binary operation symbols . . . . .	22
3	Relation symbols . . . . .	22
4	Large delimiters . . . . .	23
5	Delimiters . . . . .	23

6	Arrow symbols . . . . .	23
7	Miscellaneous symbols . . . . .	23
8	Log-like symbols . . . . .	24
9	Math mode accents . . . . .	24
10	Variable-sized symbols . . . . .	24
11	L <sup>A</sup> T <sub>E</sub> X math constructs . . . . .	24
12	AMS miscellaneous symbols . . . . .	24
13	AMS Greek and Hebrew . . . . .	25
14	AMS delimiters . . . . .	25
15	AMS arrows . . . . .	25
16	AMS negated arrows . . . . .	25
17	AMS binary operators . . . . .	25
18	AMS binary relations . . . . .	26
19	AMS negated binary relations . . . . .	26

## 1 개요

본 문서는 L<sup>A</sup>T<sub>E</sub>X<sup>1</sup> 문서 준비시스템[1]을 이용하기위해서 알아야할 내용 알려주고자 한다. 기본적인 내용만 언급 하였으며, 상세한 내용은 대부분 생략 하였다. 사실 이정도 분량의 문서로 필요한 모든사항을 포함할수는 없으며, 프로그램을 본격적으로 사용하기 위해서는 보다 완전한 참고서를 참고 하여야 한다. 이 책자의 내용만 가지고 복잡한 문서를 작성 한다면, 보다 더 많은 노력을 들이고도 만족스럽지 못한 결과를 얻게 될 것이다.

가장 널리 이용되는 참고도서는 Leslie Lamport[1]가 쓴 *The L<sup>A</sup>T<sub>E</sub>X User's guide and Reference Manual*이다. 이 책은 L<sup>A</sup>T<sub>E</sub>X사용자들이 알아야할 모든 내용을 담고 있으며, L<sup>A</sup>T<sub>E</sub>X을 사용하기 위해서는 한권쯤 갖춰야 할것이다.

메뉴얼이나 이 문서에서는 사용자의 특정 컴퓨터 시스템과 관련된 문제는 언급 하지 않는다. L<sup>A</sup>T<sub>E</sub>X은 많은 컴퓨터 시스템에서 사용 할수 있고, 그것들은 한·두가지씩 모두 다르기 때문이다. 특정 시스템에 대한것은 *local guide*를 참조 하라.

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

L<sup>A</sup>T<sub>E</sub>X을 사용하기 위해서는, 사용자는 조판할 본문과 본문을 사용자가 원하는 모양이 어떤것인지를 L<sup>A</sup>T<sub>E</sub>X에게 알려주는 지시문을 모두 담고있는 화일을 작성 하여야 한다. 이 화일은 보통 시스템 에디터를 이용하여 작성 한다. 화일의 이름은 필요한대로 정하되 내용물을 식별할수 있도록 “.tex”로 끝나야 한다. 이제 화일을 L<sup>A</sup>T<sub>E</sub>X으로 처리하면, 같은 화일이름에 확장자가 “.tex”에서 “.dvi”로 바뀐 새로운 화일이 생겨난다. 이것은 ‘Device Independent’의 약자로서, 이름이 의미하는 바 대로 모든 출력장치에 대한 출력물을 작성 할수 있다. 자세한 내용은 *local guide*를 참조 하라.

L<sup>A</sup>T<sub>E</sub>X에서 사용자는 문서가 어떻게 배치 되어야 할지를 일일이 지시 하는것이 아니라 문서의 논리적 구조를 기술하면 된다. 예를 들자면, 본문중에서 강조되는 인용문을 이러한 논리적 구조의 한 요소로 생각 할수있다. 이러한 인용문은 기본 본문과 구별될수 있도록 적절한 모양으로 조판 하는것이 일반적이다. 인간인 조판공은 인용문을 식별하고 그에 따라 취급하지만, L<sup>A</sup>T<sub>E</sub>X은 단순히 프로그램에 불과 하므로 사용자의 도움을 필요로 한다. 따라서 사용자가 인용문임을 선언하고 L<sup>A</sup>T<sub>E</sub>X이 그에 따라 인용문을 정확히 조판 하도록 해주는 L<sup>A</sup>T<sub>E</sub>X명령어가 존재 하는것 이다.

L<sup>A</sup>T<sub>E</sub>X의 기초는 문서가 정확히 어떻게 조판될것인지를 결정하는 *document class*개념이다. L<sup>A</sup>T<sub>E</sub>X은 인용문과 같은 공통적인 논리구조를 어떻게 조판할 것인지를 기술한 표준 *document class*를 제공한다. 사용자는 수학기공식과 같이 사용자 문서에 특정된 논리구조 형식을 지정 함으로서 이러한 스타일을 보충 할수 있다. 비록 완전히 새로운 스타일을 작성 하기에 앞서 조판에 대한 기본적인 원칙들을 알고 있어야 하겠지만, 표준 *document class*를 수정 하거나 심지어는 완전히 새롭게 작성하는것도 가능하다.

문서의 외형이 아니라 논리적 구조에 집중 해야할 이유는 많이 있다. 그렇게 함으로서 문서의 심미적인 질을 높인다는 잘못된 믿음—문서 디자인의 첫째 기능은 아름다움이 아니라 읽기 쉬운 문서를 만드는데 있다는점을 알아야 한다.—에서 오는 기본적인 조판 실수를 방지 할수있다. 또한 문서내의 모든 인용문의 모양을 변경 하고자 할때에도 인용문 스타일의 정의 만을 변경 하면 되므로 보다 유연하다. 가장 중요한 점은 논리적 디자인은 보다 좋은 글을 쓸수 있도록 해준다는 점이다. 비주얼 시스템은 밀집된 시스템에 비해 시각적 효과를 손쉽게 작성 할수 있다. 논리적 디자인은 글쓰기에 좀더 집중할수 있도록 해주어 좋은글을 형식과 바꾸기 힘들도록 한다.

<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

```
1: % SMALL.TEX -- Released 5 July 1985
2: % USE THIS FILE AS A MODEL FOR MAKING YOUR OWN LaTeX
3: % INPUT FILE. EVERYTHING TO THE RIGHT OF A % IS A
4: % REMARK TO YOU AND IS IGNORED BY LaTeX.
5: %
6: % WARNING! DO NOT TYPE ANY OF THE FOLLOWING 10 CHARACTERS
7: % EXCEPT AS DIRECTED:      & $ # % _ { } ^ ~ \
8:
9: \documentclass[11pt]{article}% YOUR INPUT FILE MUST CONTAIN
10: \begin{document}           % THESE TWO LINES PLUS
11:                           % THE \end COMMAND AT THE END
12:
13: \section{Simple Text}      % THIS COMMAND MAKES A SECTION TITLE.
14:
15: Words are separated by one or more spaces. Paragraphs are
16: separated by one or more blank lines. The output is not affected
17: by adding extra spaces or extra blank lines to the input file.
18:
19:
20: Double quotes are typed like this: ``quoted text''.
21: Single quotes are typed like this: 'single-quoted text'.
22:
23: Long dashes are typed as three dash characters---like this.
24:
25: Italic text is typed like this: \emph{this is italic text}.
26: Bold text is typed like this: \textbf{this is bold text}.
27:
28: \subsection{A Warning or Two} % THIS MAKES A SUBSECTION TITLE.
29:
30: If you get too much space after a mid-sentence period---abbreviations
31: like etc.\ are the common culprits)---then type a backslash followed by
32: a space after the period, as in this sentence.
33:
34: Remember, don't type the 10 special characters (such as dollar sign and
35: backslash) except as directed! The following seven are printed by
36: typing a backslash in front of them: \$ \& \# \% \_ \{ and \}.
37: The manual tells how to make other symbols.
38:
39: \end{document} % THE INPUT FILE ENDS LIKE THIS
```

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.

Italic text is typed like this: *this is italic text*. Bold text is typed like this: **this is bold text**.

### 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 화일의 처리결과

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

그림 1을 살펴보자, 이것은 표준 L<sup>A</sup>T<sub>E</sub>X 예제 화일 `small.tex`을 약간 변형한 것이다. 이러한 화일의 복사본 작성에 관해서는 `local guide`를 참조하라. 좌측에 있는 행번호는 화일의 일부는 아니지만, 참조를 손쉽게 하기 위하여 첨가 하였다. 이 화일의 처리 결과가 그림 2에 있는 그대로 나타나 있다.

### 3.1 Running Text

모든 문서는 거의 대부분이 *running text*—단어로 이루어진 문장과, 그 문장으로 이루어진 단락들—로 구성되어 있고, 이점에 있어서는 예제화일도 예외가 아니다. *Running text*를 기술 하는데는 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 단어간의 공백을 조정하고 행을 가득 채운다. 입력화일에서 공백을 주거나 단어를 띄어두는 것은 최종 출력물에 아무런 영향도 미치지 못한다. 입력화일에 들어있는 공백이 몇 개이든지간에 L<sup>A</sup>T<sub>E</sub>X은 이를 하나의 공백으로 처리한다. 또한 L<sup>A</sup>T<sub>E</sub>X은 각행의 끝을 단어간의 공백으로 간주 한다 (15–17행 참조). 입력화일에서 공백인 행은 새로운 단락의 시작임을 가르키므로, 새로운 단락을 시작하는것이 아니라면 입력 화일에 공백인 행을 두지 말아야 한다.

L<sup>A</sup>T<sub>E</sub>X은 잘 쓰이지 않는 키보드 문자를 자신의 사용을 위해 확보 하고 있다. 사용자의 입력화일에 이와 같은 열개의 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

입력화일에는 ‘\’로 시작되는 단어들이 있다 (9, 10 및 13행 참조). 이들은 L<sup>A</sup>T<sub>E</sub>X 명령어로

서 사용자 문서의 구조를 기술하는데 사용된다. 이러한 명령어의 특징은 다음과 같다.

- 모든 L<sup>A</sup>T<sub>E</sub>X 명령어는 ‘\’로 시작하고 하나 또는 그 이상의 문자로 이루어 진다.
- L<sup>A</sup>T<sub>E</sub>X 명령어는 대·소문자를 구분 한다. 따라서 \BEGIN과 \Begin 그리고 \begin은 각각 서로 다르다.
- 몇몇 명령어들은 사용자의 본문중에 위치 하기도 한다. 이러한 명령어들은 다른 글꼴과 같은것을 선택하는 스위치와 같은 역할을 한다. \Small.tex에서는 본문의 강조—보통 *이탤릭체* 글꼴로 바꾸어 표현한다—를 위해 이를 사용하고 있다 (25행 참조). 명령어와 본문은 항상 ‘{’와 ‘}’로 둘러싸여 있으며, 명령어 ‘{\em}’은 기능을 활성화 하고 ‘}’를 이를 비활성화 한다. 이와 같은 글꼴 변경 명령어는 모두 매개변수를 취하는 명령어형을 별도로 가지고 있다. 자세한것은 10 페이지를 참조하라.
- 다음과 같은 형태의 명령어도 있다.

```
\command{text}
```

이때, text를 명령어의 매개변수라고 한다. 명령어 \section이 이와 같은 것이다 (13행 참조). 매개변수는 중괄호 ‘{ }’나 각괄호 ‘[ ]’속에 표기한다. 경우에 따라서 두 가지 모두를 사용하기도 한다. 명령어는 있는 그대로 정확하게 사용 하여야 한다.

- 명령어가 문자로만 이루어져 있을때 예는 문자가 아닌것이 명령어의 끝에 오도록 해야 한다. 명령어의 매개변수를 담고있는 중괄호나 공백을 사용하는것이 일반적이다. 공백을 사용한 경우에 L<sup>A</sup>T<sub>E</sub>X은 그 공백을 무시한다. 이로 인하여 발생할수 있는 문제에 대해서는 뒤에서 살펴 보겠다.

### 3.3 전체적인 구조

모든 문서에 반드시 사용해야 하는 L<sup>A</sup>T<sub>E</sub>X명령어가 있다. 실제 본문은 \begin{document} 명령어로 시작하고 \end{document} 명령어로 끝난다 (10 and 39행 참조). \end{document} 이후에 있는 모든것은 무시된다. \begin{document} 이전에 나타 나는것 들을 서두(preamble)라고 한다. 서두(preamble)에는 문서의 형식을 기술하는 명령어만 사용하여야 한다.

서두(preamble)에서 반드시 사용하여야 하는 명령어가 \documentclass 이다 (9행 참조). 이 명령어는 문서의 전체적인 형식을 지정 한다. 예제 화일은 간단한 기술문서 이므로 article 형식을 사용하고, 11 포인터 글꼴로 인쇄 하도록 수정 하였다. 제 4 절에 나타나 있는것과 같은 다른 형식의 사용도 가능 하다.

### 3.4 기타 살펴볼 사항

L<sup>A</sup>T<sub>E</sub>X은 열고 닫는 따옴표와 쌍따옴표를 모두 식자 할수 있다. 이는 키보드에 있는 두개의 따옴표—‘ (grave accent 또는 back-quote와 유사하다) 와 ’ (apostrophe)—를 이용한것 이다. 따옴표 에는 한번 (21행 참조), 쌍따옴표 에는 두번을 사용한다(20행 참조). 쌍따옴표 문자 자체 "는 거의 사용하지 않는다.

L<sup>A</sup>T<sub>E</sub>X은 서로 다른 세가지의 대쉬를 사용한다. 문장에서 기호로 사용하는 긴 대쉬는 세개의 대쉬 ‘---’를 연속해서 입력 한다. 숫자의 범위 표시에 사용하는 짧은 대쉬 ‘10–20’는 두개의 대쉬를 연속해서 입력하고 한개의 대쉬는 하이픈으로 사용 한다.

때때로 본문에 L<sup>A</sup>T<sub>E</sub>X에서 사용하는 특수 문자를 입력해야할 필요가 있다. 그중 일곱개의 문자는 앞에다 역 슬래쉬를 붙여 입력할수 있다 (36행 참조). 나머지 세개의 문자는 †, ‡, §, £, ©, ‡ 그리고 ♣ 와 같이 키보드에 없는 문자를 입력 하는것과 마찬가지로 다른 명령어를 이용해 작성 한다.

자신이 작업한 내용이나 이유등을 알수 있도록 L<sup>A</sup>T<sub>E</sub>X화일에 주석을 첨가해 둔다면 매우 유용할 것이다. L<sup>A</sup>T<sub>E</sub>X은 % 기호 우측에 있는 모든것을 무시 하므로 이를 주석문으로 사용한다.

## 4 도큐먼트 클래스와 옵션

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

**article** 은 짧은 문서나 인쇄할 기사와 같은것에 어울린다.

article은 chapter가 없고 \maketitle로 제목을 작성 ( 제 9 절 참조)하면, 표지를 별도로 사용하지 않고 첫 페이지의 상단에 제목을 인쇄 한다.

**report** 는 좀더 기술관계의 문서작성에 알맞다. 이는 chapter를 가지며 제목을 별도의 페이지에 인쇄하는점을 제외하면 article과 같다.

**book** 은 책자의 출판을 목적으로 한다. 종이의 양면에 인쇄 할것으로 가정하여 페이지의 구성을 조정 한다.

**letter** 는 편지등을 작성하기 위한것 이다. 편지 형식을 사용하면 주소, 날자, 서명, etc. 등의 각 구성요소가 적절하게 배치된 편지를 작성 할수 있다.

이러한 표준 형식을 변경할수 있는 옵션이 다수 존재 한다. 명령어 \documentclass 다음의 각괄호 안에 들어 있는것이 바로 그것이다. 문서에서 클래스는 오직 하나만이 사용가능한 반면에, 옵션은 옵션명을 콤마로 구분하여 여러개를 동시에 사용할수 있다. 표준 옵션은 다음과 같다.

**11pt** 본문의 내용을 일반적으로 사용되는 10 포인트가 아닌 11 포인트 글꼴로 로 식자 한다. 11 포인트는 10 포인트에 비해 약 10 퍼센트 가량 크다.

**12pt** 본문의 내용을 일반적으로 사용되는 10 포인트가 아닌 12 포인트 글꼴로 로 식자 한다. 12 포인트는 10 포인트에 비해 약 20 퍼센트 가량 크다.

**twoside** article 이나 report 형식에서 문서를 종이의 양면에 인쇄 하도록 한다. 이는 book 형식 에서는 기본값이다.

**twocolumn** 각 페이지에 본문을 2단으로 조판 한다.

**titlepage** article 형식에서 \maketitle 명령어가 새로운 페이지에 표지를 작성 하도록 한다. report와 book 형식 에서는 표지가 기본적으로 새로운 페이지에 작성 된다.

유럽 국가에 유용한 옵션이 있다. (L)(v)옵션 a4paper은 모든 표준 형식의 출력을 A4 용지에 맞게끔 조정 한다. (L<sup>A</sup>T<sub>E</sub>X이 만들어진 미국에서는 A4 용지 보다 길이가 조금 짧고 폭이 넓은 용지가 표준 이어서, 이 옵션 없이 인쇄 하면 어딘지 모르게 어색 할것 이다.)

## 5 환경

우리는 앞에서 L<sup>A</sup>T<sub>E</sub>X이 제대로 조판 할수 있도록 인용문을 식별하는 것에 대해 언급 하였다. 인용문은 \begin{quotation} 와 \end{quotation} 명령어로 둘러싸서 이것이 인용문임을 식별한다. 이것은 환경이라고 불리는 L<sup>A</sup>T<sub>E</sub>X 구조의 한 예 이다. 본문을 특정 환경에 배치하면 여러가지 특수한 효과를 얻을수 있다.



## 5.1 인용

인용에는 `quote`와 `quotation`의 두가지 환경이 있다. `quote`는 짧은 인용문이나 공백인 행으로 분리된 짧은 인용문이 연속 될때 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 한단락을 넘어설 경우에는 `quotation`환경을 사용한다. 일반적으로 입력에서 단락은 공백인 행으로 구분 한다.

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 ap-  
plications 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}
```

## 5.2 중앙및 좌·우 정렬

`center` 환경에 배치된 본문은 페이지의 중앙에 정렬 되고, `flushleft`나 `flushright` 환경에 배치된 본문은 좌·우측 끝에 정렬 된다. `center`의 철자에 주의 하라 — 불행히도 L<sup>A</sup>T<sub>E</sub>X은 영국식 철자(centre)를 받아 들이지 못한다.

이와같은 환경속에 있는 본문은 일반적인 방식으로 조판된다, 특히 사용자가 입력한 행의 끝 문자는 단순히 공백으로 간주 된다. 새로운 행임을 알리기 위해서는, `\\`명령어를 입력 하여야 한다. 예를 들어 보자.

one two three

four

five

```
\begin{center}
```

one

two

three \\

four \\

five

```
\end{center}
```

## 5.3 리스트

리스트를 작성하기 위한 세가지 환경이 존재한다. 각각의 환경에서 새로운 항목은 `\item`명령어로 시작한다. `enumerate`환경에서는 각 항목이 번호로 표시 되는 반면에 `itemize`환

경에서는 특정한 마크로 표시된다. 이 환경들은 서로 내포 될수 있으며, 이경우 들여쓰기의 양과 마크는 적절히 조정 된다.

- 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}
```

리스트를 작성하기 위한 세번째 환경이 `description` 이다. `description` 환경에서는 `\item` 명령어 뒤의 각괄호에 항목의 레이블을 지정 할수 있다. 예를 들어 보자.

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 원형 출력

경우에 따라서는 터미널 화면에 나타난 모양대로 본문을 입력할 필요가 있다. 컴퓨터 프로그램의 입력과 같은것이 그 예가 될것이다. 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 영향을 주지 않고 키보드에 있는 모든 문자들을 사용하는것도 포함 될것 이다. `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}
```

## 6 글꼴

우리는 이미 강조 글꼴을 사용하기 위해 `\emph` 명령어를 사용 했었다. 글꼴을 변경하기 위한 명령어의 전체 리스트가 여기 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 <b>sans serif</b> 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 *한쌍의 괄호 안에서* 사용된다. 다른 모양으로 조판 하고자 하는 본문을 매개변수로 하는 명령형(첫번째 열)의 의 사용을 권한다. 글꼴 명령에 더하여, 글꼴의 크기를 변경하는 명령어도 있다.

tiny	scriptsize	footnotesize	small	normalsize	large	Large	<code>\tiny</code>	tiny
							<code>\scriptsize</code>	scriptsize
							<code>\footnotesize</code>	footnotesize
							<code>\small</code>	small
							<code>\normalsize</code>	normalsize
							<code>\large</code>	large
							<code>\Large</code>	Large
							<code>\LARGE</code>	LARGE
							<code>\huge</code>	huge
							<code>\Huge</code>	Huge

## 7 섹션 명령과 목차

이문서와 같은 기술관계 문서들은 보통 여러 섹션으로 구분 된다. 각각의 섹션은 참조를 보다 쉽게 하기 위해서 제목과 번호를 부여 한다. L<sup>A</sup>T<sub>E</sub>X에는 각기 다른 종류의 섹션을 식별하기 위한 일련의 명령어가 준비 되어 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>

위의 두명령어는 명칭이 적절하지 못하다, 일반적인 단어의 의미대로 단락과 관련된 것이 아니라 이는 단지 section의 하위 레벨 일 뿐이다. 대부분의 문서에서 `\paragraph`와 `\subparagraph`에서 만들어진 제목에는 번호를 부여하지 않는다. article 형식에서는 `\chapter`를 사용할수 없다. 각 명령어는 section 은 chapter 내에서, subsection은 section 내에서 사용 하는것과 같이 주어진 순서대로 사용 하여야 한다.

일곱번째 섹션 명령인 `\part`도 있다. 이 명령어의 사용은 선택적 으로서, 이는 대용량의 문서를 일련의 부분으로 구분 하는데 사용된다. 이 명령어는 다른 명령어에서 부여한

번호를 그대로 보전 한다.

사용자는 문서에서 `\tableofcontents`를 사용하여 각종 섹션 명령을 바탕으로 하여 작성된 내용에 따른 목차를 작성·포함 할수 있다. 문서를 매번 L<sup>A</sup>T<sub>E</sub>X으로 처리할때마다 직전에 L<sup>A</sup>T<sub>E</sub>X 처리에서 얻어진 제목등으로 목차가 새롭게 구성 된다는점에 유의 하여야 한다. 이는 L<sup>A</sup>T<sub>E</sub>X 문서를 처리할때 목차에 필요한 정보를 모아 두었다가 다시 문서를 처리 할때 이 정보를 이용하기 때문이다. 이때문에 정확한 목차를 작성하기 위해서 문서를 L<sup>A</sup>T<sub>E</sub>X으로 2번 처리 하여야 할 경우도 있다.

## 8 특수기호의 식자

사용자는 키보드에는 없는 다양한 종류의 특수기호들을 자신의 문서에 나타낼수 있다. 먼저 어떠한 글자에도 엑센트를 표기할수 있다.

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

기타의 특수기호와 그 입력 명령어를 아래에 표시 하였다.

†	<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>	l	<code>\l</code>
Ł	<code>\E</code>	ß	<code>\ss</code>	¿	<code>?'</code>
ı	<code>!'</code>	...	<code>\ldots</code>	L <sup>A</sup> T <sub>E</sub> X	<code>\LaTeX</code>

명령어 `\today`는 현재 날짜를 문서에 삽입한다. 이러한 L<sup>A</sup>T<sub>E</sub>X 명령어를 사용 할 때에는 L<sup>A</sup>T<sub>E</sub>X이 명령어뒤의 모든 공백을 무시한다는 점에 유의 해야 한다, 따라서 `'\pounds 20'`라고 입력 하면 `'£20'`와 같이 나타난다. 어쨌든 `'\LaTeX is wonderful'`라고 입력하면 `'LATEXis wonderful'`이라고 나타나는데, L<sup>A</sup>T<sub>E</sub>X 다음의 공백이 없다것에 주목하라. 이를 방지 하기 위해서는 명령어 다음에 한쌍의 빈 중괄호를 입력한 뒤에 공백을 입력 하면 된다. 이제 `\LaTeX{} really is wonderful!`는 L<sup>A</sup>T<sub>E</sub>X really is wonderful!과 같이 나타날 것이다.

마지막으로 보통 수식의 조판에 사용하는 L<sup>A</sup>T<sub>E</sub>X의 수식 모드에서는, 대·소문자의 허랍 어 수학 알파벳과 멋진 글꼴 그리고 수학 연산자와 관계 연산자 및 화살표와 기타 많은것을 포함하여 훨씬더 많은 기호를 사용할수 있다. 이에 관해서는 제 13 절에서 자세히 다룬다.

## 9 제목

대부분의 문서에는 제목이 있다. L<sup>A</sup>T<sub>E</sub>X문서에 제목을 주기 위해서는, 보통 `\begin{document}` 명령어 뒤에 다음의 명령어를 사용한다.

```
\title{문서의 제목}
\author{문서의 작성자}
\date{필요한 날짜}
\maketitle
```

문서의 작성자가 여럿일 경우에는 `\author` 명령어에 각 작성자의 이름을 `\and`로 구분하여 표시하면 각자의 이름이 서로 다른 행에 중앙정렬 되어 인쇄 된다. `\date` 명령을 사용하지 않으면 현재의 날자가 인쇄 된다.

```

\title{Essential \LaTeX}
\author{Jon Warbrick \and A N Other}
\date{14th February 1988}
\maketitle

```

Essential L<sup>A</sup>T<sub>E</sub>X  
J Warbrick   A N Other  
14th February 1988

제목의 정확한 모양은 도큐먼트 형식에 따른다. `report`와 `book` 형식에서는 표지 페이지에 인쇄 되고, `article` 형식에서는 첫번째 페이지의 상단에 인쇄 된다. 클래스 옵션인 `titlepage`를 이용하면 이를 변경 할수있다 (제 4 절 참조).

## 10 표의 사용

L<sup>A</sup>T<sub>E</sub>X은 연속된 공백을 하나의 공백으로 처리하기 때문에 표를 조판 하기가 까다로운 편이다. 어떻게 나타나는지 예제를 살펴보자.

Income Expenditure Result	<code>\begin{flushleft}</code>
20s 0d 19s 11d happiness	Income   Expenditure   Result   \
20s 0d 20s 1d misery	20s 0d   19s 11d     happiness \
	20s 0d   20s 1d     misery
	<code>\end{flushleft}</code>

`tabbing` 환경을 이용하면 이 문제를 해결 할수 있다. `tabbing` 환경에서는 타이프라이터 에서와 같이 탭스톱을 설정하여 탭으로 이동 할수 있다. 탭스톱은 `\=` 명령으로 설정 하고, `\>` 명령은 다음 탭으로 이동 한다. 줄 바꿈은 `\\` 명령으로 이루어 진다. `\kill`로 표기한 부분은 아무것도 출력 되지 않으므로 탭스톱의 설정에 이용된다.

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

타이프라이터 에서의 탭키와는 달리, `\>` 명령은 비록 그 위치가 왼쪽이라 해도 순서대로 다음 탭으로 이동 한다. 이 때문에 두 탭스톱간의 간격이 너무 적을 경우에는 글자가 겹쳐서 쓰이는 현상이 나타난다.

좀더 복잡한 표작업을 한다면, `tabular` 환경이 필요 할 것이다. 여기서는 L<sup>A</sup>T<sub>E</sub>X이 전체 표를 검토하여 가장 긴 항목을 집어 넣기 위해서는 열의 간격을 어떻게 해야 할지를 결정 하기 때문에 `tabbing`에서와 같이 열의 넓이에 대해 염려하지 않아도 된다. 사용자는 단지 몇개의 열이 있고 각 열을 어떻게 배치하고 싶은지를 L<sup>A</sup>T<sub>E</sub>X에게 알려 주기만 하면 된다. 이는 `\begin{tabular}` 명령 다음에 각 열에 대해 한 문자로 된 *template*를 이용하여 지정 한다.

- l   means the column will be left justified
- r   means the column will be right justified
- c   means the column will be centred

한 행 내에서의 항목 구분은 & 문자로 이루어지며(왜 특별히 취급 해야 하는지 이제 알 수 있을 것 이다), 한행의 끝은 \\로 표시 된다. 3열 2행으로 된 간단한 표를 작성 해보면 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 기능이 있다.

1. \hline을 사용하면 표의 행과 행 사이에 줄을 그을 수 있다.
2. *template*에 |를 사용하면 열과 열 사이에 선을 그을 수 있다.
3. *template*에 p를 사용하고 {와 } 사이에 폭을 지정 함으로서 열의 폭을 고정 할 수 있다.  
이 때 폭은 'cm', 'mm' 또는 'in'으로 표기 할 수 있다 ('inches'를 사용 할 수는 없다).

좀더 세밀한 표를 작성해 보자.

<i>Group</i>	<i>Type</i>	<i>Sherds</i>
Groups 1–9	Grey wares	218
Groups 40–44	Black (mostly 'black burnished') wares	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 & \em Sherds \\
\hline
Groups 1--9 & Grey wares & 218 \\
Groups 40--44 & Black (mostly  
'black burnished') wares & 116 \\
Groups 61--67 & Buff-red-orange  
wares & 46 \\
Groups 81--85 & Colour-coated  
fine wares & 67 \\
Groups 91--2, 93--4 & Mortaria  
and miscellaneous & 35 \\
Group 96 & Samian & 56 \\
& & 538 \\
\hline
\end{tabular}

```

사용자가 입력한 행의 끝이 중요한게 아니라는데 주목 하라, 행의 끝은 반드시 \\로 이루어 진다.

## 11 표와 그림

표에 문자와 숫자를 배치함에 있어서, 서로 다른 페이지에 나뉘어 배치 하는게 아니라 거의 항상 전체 표를 함께 배치 하여야 할 것이다. 페이지의 아랫 부분에서 충분한 공간이 확보 되지 못할 경우에는 어떻게 할 것인가? 이와 비슷한 문제로서, 나중에 사진을 붙여 넣기 위한 공간을 어떻게 확보 할 것인가? 사용자가 본문을 입력 할 당시에는 L<sup>A</sup>T<sub>E</sub>X이 어디에서 새로운 페이지를 시작할지 알 수 없으므로 필요한 만큼의 공간을 확보 하기가 매우 어렵다. L<sup>A</sup>T<sub>E</sub>X은 정확히 사용자의 입력 화일에 있는 그 위치가 아니라 페이지내의 적당한 위치에다 이를 배치하는 이동개체(float object) 시스템을 이용하여 이 문제를 해결 하고 있다. 보통 현재 페이지의 하단이나 다음 페이지의 상단 또는 더 많은 공간이 필요 하다면 개체만의 페이지에 배치한다. L<sup>A</sup>T<sub>E</sub>X은 두 종류의 개체를 지원 하는데, *table*과 *figure*라고 하는 것

이 그것으로서, 이들은 각기 캡션을 지정 할수 있다. 그림을 표시하기 위해 3 인치의 공간을 확보하고 캡션을 지정하는 예는 다음과 같다.

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

표와 그림의 번호는 섹션에서와 같이 자동으로 붙여진다. `\tableofcontents` 명령이 각 섹션과 서브섹션등의 목록을 페이지 번호와 함께 표시해 주듯이 `\listoffigures`과 `\listoftables` 명령은 그림과 표의 목록을 캡션과 함께 표시해 준다.

## 12 상호참조와 인용

L<sup>A</sup>T<sub>E</sub>X의 가장 유용한 기능중의 하나가 상호 참조에 사용될 수식, 표, 그림, 페이지, 섹션 등의 번호를 자동으로 생성해 주는것 일것 이다. 예를 들어, 도큐먼트 스타일에 대해서는 제 4 절을 참조하고, `sample.tex`의 처리결과는 그림 2을 참조 하는것과 같은것은

도큐먼트 스타일에 대해서는 제~\ref{sec:styles}~절을 참조 하고,  
`\fn{sample.tex}`의 처리결과는 그림~\ref{fig:result}을 참조

와 같이 표시된다. 문서에서 참조번호가 인쇄된 곳에는 번호 대신에 `\ref` 명령어가 있을 수 있다. ~는 L<sup>A</sup>T<sub>E</sub>X이 줄바꿈을 할수 없는 공백을 삽입한다.

상호 참조를 하기 위해서는 L<sup>A</sup>T<sub>E</sub>X 입력 화일에서 참조될 부분도 당연히 표시되어야 한다. 이는 `\label` 명령으로 이루어 진다. 섹션의 레이블은 다음처럼 섹션 제목 바로뒤에 지정 한다.

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

and that for a figure or table in or immediately following the `\caption` command:

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

`\label`과 `\ref`에는 어떠한 이름 일지라도 사용할수는 있지만, 위의 관례를 따르는것도 나쁘지는 않다. `\label`은 페이지 참조에도 사용된다, 다만 `\ref`가 아니라 `\pageref`를 사용한다.

L<sup>A</sup>T<sub>E</sub>X 입력화일 내에서 문서의 일부분을 기호로 참조하는것과 동일하게, 다른 문서를 인용 할 수도 있다. 문서에 대한 인용은 다음처럼 `\cite` 명령을 사용해 이루어 진다.

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

보통 *citation key*라고 하는 `\cite` 명령어의 매개변수는 *bibliographic database*에 들어있는 책자나 논문을 구분·인식 하는데 이용된다.

문서에서 다른 문서를 인용한다면, 다음의 명령어로 참조목록을 만들수 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, 문서내 에서는 각괄 호안에 표기된다.

**unsrt** 인용물의 목록은 문서에서 나타나는 순서대로 정렬되어 번호가 부여되고, 문서 내에서는 각괄호안에 표기된다.

**alpha** 인용물의 목록은 알파벳 순으로 정렬되나 번호가 아니라 “Lam86”과 같이 레이블이 붙여지고, 문서 내에서는 각괄호안에 표기된다.

**abbrev** alpha와 비슷하나 전체적으로 더 간단하다. (e.g., 잡지의 이름과 월을 줄이는 등)

명령어 `\bibliography`의 매개변수는 (i.e., 위의 예제에서 `mybib1.bib`나 `mybib2.bib`와 같이) 확장자가 `.bib`인 화일 이름을 콤마로 구분하여 표기한다. 이 화일에는 `\cite` 명령에 대한 완전한 참조가 들어 있다. 이러한 `.bib`화일의 형식이나, 인용을 어떻게 L<sup>A</sup>T<sub>E</sub>X과 호환성 있는 형식으로 변환 하느냐 하는등의 문제는 이 안내서의 주제를 넘어서는 것이다. 부록 B에 있는 Lamport의 책에 자세히 설명 되어 있다.

## 13 수식의 조판

### 13.1 Math, Display-math 그리고 Equation

T<sub>E</sub>X은 수식을 일반적인 텍스트와는 완전히 달리 취급한다. 수식을 위해서는 *math mode*와 *display math mode*라고 알려진 두가지의 모드가 존재한다.

Math mode는 `\(...\)`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.
```

Display math mode 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.

```
\[ f(x) = \sum_{i=0}^{\infty}
\frac{f^{(i)}(x)}{i!} \]
```

Display math mode의 변형인 `equation`은 자동적으로 수식에 번호를 붙인다.

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

```
(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 mode`에서 사용하는 주요 명령어를 보여주고 있다. 다음과 같은 점에 주목 하여야 한다.

1. 지수와 첨자는 `_`와 `^`로 표기한다. 예를들어 `x_{1} = p^{2}`는  $x_1 = p^2$ 와 같이 나타난다.
2. 분수는 `\frac`명령으로 표시한다. 예를들어 `\(\frac{a + b}{c}\)`는  $\frac{a+b}{c}$ 로 나타난다.
3. 수학기호를 나타내는 많은 명령이 있다.  
`\infty \rightarrow \sqrt{\bigotimes}` 은  $\infty \Rightarrow \sqrt{\otimes}$  와 같이 표시된다.
4. 배열은 `array`환경으로 표현한다. 이는 각 항목이 `LR mode`가 아니라 `math mode`로 조판된다는 점을 제외 하면 제 10 절 에서 설명한 `tabular` 환경과 동일하다. `array`환경에서는 배열에 괄호를 표시 하지 않는데 주목 하여야 한다. 따라서 행렬식이나 심지어는 열을 중심으로 정렬 하고자 하는 일련의 수식에도 이를 이용할수 있다.
5. 명령어 `\left`와 `\right`를 사용하면 필요한 구분자를 적절한 크기로 조판 할수 있다. 구분자로 사용할수 있는 많은 기호가 있다, e.g., `\left( \left\{ \left| \right. \right.`. 구분자 전체 목록을 아래의 표 5와 표 4에 나타 내었다.

## 13.2 간격

`Math mode`에서는 입력화일에 있는 모든 간격을 무시한다. 간격을 조정 하고자 할때에는 당므의 명령을 사용한다.

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

L<sup>A</sup>T<sub>E</sub>X에서 간격 조정이 필요한 좋은 예가 있다.

$\iint z \, dx \, dy .. \int \int z \, dx \, dy$                       `\(\int\!\!\!\int z\,, \, dx \, dy .. \int\!\int z \, dx \, dy\)`

## 13.3 수식모드에서 글꼴의 변경

수식모드의 기본 글꼴은 *math italic* 이다. 이를 일반적인 *text italic*과 혼돈 해서는 안된다. 일반적인 문자의 글꼴은 일반적인 명령인 `\emph`, `\textbf`, etc. 등으로 변경할수 있다. 히랍어 소문자 (`\alpha`등)는 수학기호로 간주되고 (이는 수식모드에서 입력해야 한다는 것을 의미한다), 이러한 글꼴 변경 명령에 영향을 받지 않는다.

명령어 `\mathbf`는 **bold face roman**문자를 식자 한다. 볼드체 히랍어나 수학기호 및 ***bold face math italic***을 사용 하고자 한다면 수식모드에 들어가기전에 `\boldmath`명령을 사용하라. 그럼 수식모드의 기본 글꼴이 볼드로 변경된다.

$x = 2\pi \Rightarrow x \simeq 6.28 \, x = 2\pi \Rightarrow x \simeq 6.28 \, x = \mathbf{2\pi} \Rightarrow x \simeq$       `\( x = 2\pi \rightarrow x \simeq 6.28 \)`  
**6.28**                      `\(\mathbf{x} = 2\pi \rightarrow x \simeq 6.28 \)`  
                              `\{\mathbf{x} = \mathbf{2}\pi \rightarrow x`  
    `\simeq\{\mathbf{6.28}\}`                      `\}`

대문자에 대해서는 `calligraphic` 글꼴이 존재 한다. 이는 `\mathcal`명령으로 생성한다.

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

## 14 \$가 의미 하는 것은?

$\mathcal{A}\mathcal{M}\mathcal{S}$ -T<sub>E</sub>X이나 plain T<sub>E</sub>X을 사용하다가 L<sup>A</sup>T<sub>E</sub>X으로 전환 하였다면, 왜 \$나 \$\$에 대해 언급이 없는지 궁금 할것이다.

이 시스템들은 math mode는 \$로 둘러싸서 표시하고, display math mode는 \$\$로 둘러싸서 표시한다. 이는 문서 편집기에서 \$기호의 짝을 맞추기가 어렵고 math mode의 시작과 끝이 어딘지를 알기가 어려우므로 L<sup>A</sup>T<sub>E</sub>X 시스템에 비해 단점이 된다. 사용자가 \$ 기호를 빠뜨리면 T<sub>E</sub>X도 혼란 스럽게 된다.

(잘못된) 입력

```
let (a,b,c)$ be a Pythagorean 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 수식모드 바깥에서 ^명령을 사용 하였다고 하여 에러의 내용과 위치를 잘못 표기 하고 있다. T<sub>E</sub>X은 ‘be a ... such that’을 수식모드로 조판하고 ‘such that’다음의 \$에서 수식모드를 마친다.

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

```
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 \((를 빠뜨렸 으므로 수식모드가 아닌곳 에서 수식모드의 끝을 표시하는 \))를 사용 하였다고 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 that \...
```

?

달라 기호는 조그만 수식의 표현에 유용하다.

Let  $G$  be a  $p$ -group

Let  $\$G\$$  be a  $\$p\$$ -group

두개의 달라 기호가 항상 `\[ ... \]`와 동일한것은 아니므로, 다른 스타일이나 옵션에서도 호환성을 유지 하고 싶다면 이를 사용하지 말아야 한다 (*fleqn* 클래스 옵션을 사용해 보라).

## 14.1 기호

다음의 표는 표준  $\text{\LaTeX}$  기호글꼴에서 사용가능한 기호 대부분을 보여 준다. 관계연산자에 대한 부정은  $\text{\textbackslash not}$ 로 이루어 진다.

$$G \not\equiv H \qquad \qquad \$G \not\equiv H$$$

부록에 표 1에서 표 11까지  $\text{\LaTeX}$ 의 표준 기호를 종류별로 정리 해 두었다. 표 13에서 표 12에 있는  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ 의 추가적인 기호들은 `amssymb` 패키지를 지정 해야지만 사용할 수 있다.

## 15 편지

TeX으로는 간단하게 편지를 작성 할수 있다. 단지 letter형식의 도큐먼트 스타일을 사용 하기만 하면 된다. 하나의 입력파일로 여러장의 편지를 작성 할 수있다. 본인의 이름과 주소처럼 모든 편지에서 동일하게 사용되는 내용은 화일의 처음에 한번만 정의 해 준다. 각각의 편지는 수신인의 주소와 이름을 매개변수로 하는 letter 환경으로 작성 된다. 편지의 본문은 \opening명령에 따라 인사와 함께 시작 된다.

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

다음의 예제를 살펴보면 좀더 명확하게 이해 할 수 있을 것이다. An example will make this clearer:

```
\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}
```

## 16 에러

새로운 L<sup>A</sup>T<sub>E</sub>X 입력 파일을 작성 하다 보면 아마 실수를 하게 될 것이다. 누구나가 그런 실수는 하는 것 이므로 그렇게 염려 할 필요는 없다. 여기에는 다른 컴퓨터 프로그램과 같이 두 종류의 실수가 있다. L<sup>A</sup>T<sub>E</sub>X에서 경고를 해주는 것과 그렇지 않은 것 두 가지 종류이다. 간단히 예를 들자면, 사용자의 본문을 잘못 입력 했다면 L<sup>A</sup>T<sub>E</sub>X은 이를 알지 못하고 사용자는 결과물을 알아서 읽으면 된다. 반면에 L<sup>A</sup>T<sub>E</sub>X의 환경명을 잘못 입력 했다면 L<sup>A</sup>T<sub>E</sub>X은 어떻게 해야 할 지를 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. 생성된 에러 메시지는 사용자가 이해하기 쉬운 것으로 생각한다. 어찌 되었건 대충 어디에서 무엇이 잘못 되었는지를 알려 주므로 어디를 살펴 보아야 할 지 알 수 있다.

`\begin{itemize}`를 `\begin{itemie}`로 잘못 입력하면 어떻게 되는지 살펴 보도록 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 감지 하였음을 알려 주는 것 뿐이다. ‘!’로 시작하는 세 번째 줄은 *에러 지시문* 이다. L<sup>A</sup>T<sub>E</sub>X에 경험이 쌓이기 까지는 그 의미가 충분 하지는 않겠지만, 이는 어떤 문제가 일어 났는지를 알려 준다. 이 경우에는 단순히 `itemie`라는 알 수 없는 환경이 나타났음을 말하고 있는 것이다. 다음의 두 줄은 에러가 발생 하였을 때 L<sup>A</sup>T<sub>E</sub>X이 무엇을 하고 있었는지를 알려준다. 지금은 별 관계가 없는 내용 이므로 무시한다. 마지막 줄은 *에러 위치*를 나타내며, 사용자의 입력파일에서 문제가 된 줄을 복사하여 나타낸다. 사용자가 손쉽게 찾을 수 있도록 행번호로 시작 한다. 에러가 라인 중간에서 발생 하였다면, L<sup>A</sup>T<sub>E</sub>X이 에러라고 인식한 그 부분에서 행을 분리하여 표시 한다. 다른 모든 컴퓨터 프로그램과 마찬가지로 L<sup>A</sup>T<sub>E</sub>X에서도 가끔씩 잘못을 발견하기 전에 실제 에러가 있는 부분을 지나쳐 버리기도 한다, 하지만 대부분 그리 멀지는 않다.

이 경우 몇 가지 행동을 취할 수 있다. L<sup>A</sup>T<sub>E</sub>X을 잘 알고 있다면 문제를 수정 할 수도 있을 것이고, 그렇지 않다면 ‘x’를 입력하여 L<sup>A</sup>T<sub>E</sub>X의 가동을 중단 시킨 다음 에러를 수정 할 수도 있을 것이다. 제일 좋은 방법은 그냥 리턴키를 누르는 것 이다. 그러면 L<sup>A</sup>T<sub>E</sub>X은 마치 아무 일도 없었던 것 처럼 계속해서 작동 하게 된다. 하나의 실수를 저질렀다면, 다른 실수도 저질렀을 수도 있으므로 모든 에러를 한번에 찾고자 할 것이다. 그편이 에러가 있을 때 마다 매번 L<sup>A</sup>T<sub>E</sub>X을 실행해서 하나씩 에러를 수정 하는 것 보다 훨씬 효율적일 것이다. 나타나는 모든 메시지는 나중의 참고를 위해 *log* 파일에 저장 되므로 어떤 에러가 있었는지 기억할 필요는 없다. 이 파일을 무엇이라고 하는지는 각 시스템의 *local guide*를 참조 하라.

에러가 일어난 행을 살펴보면, 문제가 무엇인지 대략 알 수가 있을 것이다. 무엇이 문제인지 알 수 없을 때에는 아래의 힌트를 살펴보고, 그래도 도움이 되지 않는다면 Lamport의 책 제 6 장을 참고 하라. 접하게 되는 에러 메시지 전부를 수록 하고 있으며, 에러가 일어난 원인등에 대한 제안도 함께 담고 있다.

에러를 일으키는 가장 보편적인 실수는 다음과 같다.

- 명령어나 환경의 이름을 잘못 입력 하는 경우.

- ‘{’와 ‘}’의 짝이 맞지 않을 경우 — 항상 짝을 맞추어 사용 해야 한다.
- 특수기호 # \$ % & \_ { } ~ ^ \를 일반 문자로 그냥 사용한 경우.
- \end명령어를 빠뜨린 경우.
- 명령어의 매개변수(중괄호 사이에 있다)를 빠뜨린 경우.

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

가끔씩 L<sup>A</sup>T<sub>E</sub>X은 에러 메세지도 없이 \*를 출력 하고는 정지해 버리기도 한다. 다른것이 원인 일수도 있지만 대부분의 경우에 이것은 \end{document}명령어를 빠뜨려서 일어난 것 이다. 이런일이 일어나면 \stop이라고 입력하고 리턴키를 눌러라.

마지막으로, L<sup>A</sup>T<sub>E</sub>X은 가끔씩 *warning* 메세지를 출력 한다. 이것은 L<sup>A</sup>T<sub>E</sub>X의 작업을 중단 할것 까지는 없으나 조사해볼 필요가 있는 문제가 있음을 보고 하는것 이다. 대부분이 텍스트 라인의 ‘overfull’과 ‘underfull’ 문제이다. 다음의 메세지는

```
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 10.6포인트 만 큼 이다. 1 포인트는 1/72.27 인치 이므로, 이를 알아 보기는 어렵다. 하지만 아무것도 아닌 것은 아니다.

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

이와는 달리 “Underfull \hboxes”메세지가 나타날 수도 있다. 이는 단어와 단어 사이에 L<sup>A</sup>T<sub>E</sub>X이 원하는것 보다는 더많은 공백이 있는 행임을 의미한다. 일반적으로 이 경우에 할 수 있는 일이란 별로 없다. 행이 늘어진 느낌이 있겠지만, 출력물은 괜찮아 보일것이다. 이와 같은 경우에 할수 있는 유일한것은 문제의 단락을 재작성 하는것 이다!

## 17 마치는 글

이제 다양한 종류의 문서를 작성 할수 있을 정도로 L<sup>A</sup>T<sub>E</sub>X에 대해 많이 알수 있을것 이다. 하지만 이문서에서 다룬것은 L<sup>A</sup>T<sub>E</sub>X이 할수 있는 일에 비한다면 겨우 수박의 겉을 (ㄴ)(v)훔은 것에 지나지 않는다. 이 문서 전체를 L<sup>A</sup>T<sub>E</sub>X으로 작성 하였다 (속임수나 복사기를 사용하지 않고서). 게다가 사용할수 있는 모든 특성을 다 사용한것도 아니다. 이로 미루어 여러분의 처분에 맡겨져 있는 L<sup>A</sup>T<sub>E</sub>X의 파워를 조금 이나마 느낄수 있을것 이다.

복잡한 문서를 작성 하고자 한다면 *Lamport*의 책을 참조하라 고한 이 문서의 개요 에서 얘기한 내용을 기억 하라. 여기에서 얻은 내용에만 머무른 다면 시간만 낭비 하게 될것 이다. L<sup>A</sup>T<sub>E</sub>X을 어느정도 사용하고 나면, 문서를 조판하는 L<sup>A</sup>T<sub>E</sub>X 사용자들을 돕기 위해 개발 된 많은 기능 확장 화일들을 어떻게 사용하는지 알고 싶을것 이다. 이에 관한 보다 많은것 들 ( 색인의 작성을 위해 *MakeIndex*를 어떻게 사용하는 지, 또는 참고 문헌을 관리하기 위한 *BibT<sub>E</sub>X*을 어떻게 사용 하는지와 같은 )을 *The L<sup>A</sup>T<sub>E</sub>X Companion* [3]에서 다루고 있다. 마지막으로, T<sub>E</sub>X에 대한 궁극적인 한마디는 Knuth교수의 *T<sub>E</sub>X book* [2] 이다.

그리고 또다른 경고 한마디, L<sup>A</sup>T<sub>E</sub>X을 가지고 놀다보면 여러분의 문서가 결코 다시 예전 과 같아 질수는 없다...

## A 수학기호

$\alpha$	<code>\alpha</code>	$\beta$	<code>\beta</code>	$\gamma$	<code>\gamma</code>	$\delta$	<code>\delta</code>
$\epsilon$	<code>\epsilon</code>	$\varepsilon$	<code>\varepsilon</code>	$\zeta$	<code>\zeta</code>	$\eta$	<code>\eta</code>
$\theta$	<code>\theta</code>	$\vartheta$	<code>\vartheta</code>	$\iota$	<code>\iota</code>	$\kappa$	<code>\kappa</code>
$\lambda$	<code>\lambda</code>	$\mu$	<code>\mu</code>	$\nu$	<code>\nu</code>	$\xi$	<code>\xi</code>
$o$	<code>o</code>	$\pi$	<code>\pi</code>	$\varpi$	<code>\varpi</code>	$\rho$	<code>\rho</code>
$\varrho$	<code>\varrho</code>	$\sigma$	<code>\sigma</code>	$\varsigma$	<code>\varsigma</code>	$\tau$	<code>\tau</code>
$\upsilon$	<code>\upsilon</code>	$\phi$	<code>\phi</code>	$\varphi$	<code>\varphi</code>	$\chi$	<code>\chi</code>
$\psi$	<code>\psi</code>	$\omega$	<code>\omega</code>				
$\Gamma$	<code>\Gamma</code>	$\Delta$	<code>\Delta</code>	$\Theta$	<code>\Theta</code>	$\Lambda$	<code>\Lambda</code>
$\Xi$	<code>\Xi</code>	$\Pi$	<code>\Pi</code>	$\Sigma$	<code>\Sigma</code>	$\Upsilon$	<code>\Upsilon</code>
$\Phi$	<code>\Phi</code>	$\Psi$	<code>\Psi</code>	$\Omega$	<code>\Omega</code>		

표 1: Greek letters

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

<sup>a</sup> Not predefined in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. Use the packages `latexsym` or `amssymb`

표 2: Binary operation symbols

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

표 3: Relation symbols

$\left\{ \right.$	$\rmoustache$	$\int$	$\lmoustache$	$\right)$	$\rgroup$	$\left( \right.$	$\lggroup$
$\downarrow$	$\arrowvert$	$\parallel$	$\Arrowvert$	$\downarrow$	$\bracevert$		

⌘ 4: Large delimiters

$\uparrow$	$\uparrow$	$\Uparrow$	$\downarrow$	$\downarrow$	$\Downarrow$
$\{$	$\{$	$\}$	$\updownarrow$	$\updownarrow$	$\Updownarrow$
$\lfloor$	$\lfloor$	$\rfloor$	$\lceil$	$\lceil$	$\rceil$
$\langle$	$\langle$	$\rangle$	$/$	$/$	$\backslash$
$ $	$ $	$\parallel$	$\backslash$		

⌘ 5: Delimiters

$\leftarrow$	$\leftarrow$	$\longleftarrow$	$\uparrow$	$\uparrow$
$\Leftarrow$	$\Leftarrow$	$\Longleftarrow$	$\Uparrow$	$\Uparrow$
$\rightarrow$	$\rightarrow$	$\longrightarrow$	$\downarrow$	$\downarrow$
$\Rightarrow$	$\Rightarrow$	$\Longrightarrow$	$\Downarrow$	$\Downarrow$
$\leftrightarrow$	$\leftrightarrow$	$\longleftrightarrow$	$\updownarrow$	$\updownarrow$
$\Leftrightarrow$	$\Leftrightarrow$	$\Longleftrightarrow$	$\Updownarrow$	$\Updownarrow$
$\mapsto$	$\mapsto$	$\longmapsto$	$\nearrow$	$\nearrow$
$\hookrightarrow$	$\hookrightarrow$	$\hookrightarrow$	$\searrow$	$\searrow$
$\leftharpoonup$	$\leftharpoonup$	$\leftharpoonup$	$\swarrow$	$\swarrow$
$\leftharpoondown$	$\leftharpoondown$	$\leftharpoondown$	$\nwarrow$	$\nwarrow$

⌘ 6: Arrow symbols

$\ldots$	$\ldots$	$\cdots$	$\vdots$	$\ddots$	$\aleph$
$\prime$	$\forall$	$\forall$	$\infty$	$\hbar$	$\emptyset$
$\exists$	$\nabla$	$\nabla$	$\surd$	$\Box^a$	$\triangle$
$\Diamond^a$	$\imath$	$\imath$	$\jmath$	$\ell$	$\neg$
$\top$	$\flat$	$\flat$	$\natural$	$\sharp$	$\wp$
$\bot$	$\clubsuit$	$\clubsuit$	$\diamondsuit$	$\heartsuit$	$\spadesuit$
$\mho^a$	$\Re$	$\Re$	$\Im$	$\angle$	$\partial$

<sup>a</sup> Not predefined in L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. Use the packages latexsym or amssymb

⌘ 7: Miscellaneous symbols

$\arccos$	$\cos$	$\csc$	$\exp$	$\ker$	$\limsup$	$\min$	$\sinh$
$\arcsin$	$\cosh$	$\deg$	$\gcd$	$\lg$	$\ln$	$\Pr$	$\sup$
$\arctan$	$\cot$	$\det$	$\hom$	$\lim$	$\log$	$\sec$	$\tan$
$\arg$	$\coth$	$\dim$	$\inf$	$\liminf$	$\max$	$\sin$	$\tanh$

⌘ 8: Log-like symbols

$\hat{a}$	$\acute{a}$	$\bar{a}$	$\dot{a}$	$\breve{a}$
$\check{a}$	$\grave{a}$	$\vec{a}$	$\ddot{a}$	$\tilde{a}$

⌘ 9: Math mode accents

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

⌘ 10: Variable-sized symbols

$\widetilde{abc}$	$\widehat{abc}$
$\overleftarrow{abc}$	$\overrightarrow{abc}$
$\overline{abc}$	$\underline{abc}$
$\overbrace{abc}$	$\underbrace{abc}$
$\sqrt{abc}$	$\sqrt[n]{abc}$
$f'$	$\frac{abc}{xyz}$

⌘ 11: L<sup>A</sup>T<sub>E</sub>X math constructs

$\hbar$	$\hslash$	$\Delta$
$\nabla$	$\square$	$\lozenge$
$\textcircled{S}$	$\angle$	$\measuredangle$
$\nexists$	$\mho$	$\Finv^a$
$\oslash$	$\Bbbk^a$	$\backprime$
$\varnothing$	$\blacktriangle$	$\blacktriangledown$
$\blacksquare$	$\blacklozenge$	$\bigstar$
$\sphericalangle$	$\complement$	$\eth$
$\diagup^a$	$\diagdown^a$	

<sup>a</sup> Not defined in style amssymb, define using the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> `\DeclareMathSymbol` command

⌘ 12: AMS miscellaneous symbols



$\digamma$  `\digamma`  $\varkappa$  `\varkappa`  $\beth$  `\beth`  $\daleth$  `\daleth`  $\gimel$  `\gimel`

⌘ 13: AMS Greek and Hebrew

$\ulcorner$  `\ulcorner`  $\urcorner$  `\urcorner`  $\llcorner$  `\llcorner`  $\lrcorner$  `\lrcorner`

⌘ 14: AMS delimiters

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

⌘ 15: AMS arrows

$\nleftarrow$  `\nleftarrow`  $\nrightarrow$  `\nrightarrow`  $\nLeftarrow$  `\nLeftarrow`  
 $\nrightarrow$  `\nrightarrow`  $\nleftarrow$  `\nleftarrow`  $\nLeftarrow$  `\nLeftarrow`

⌘ 16: AMS negated arrows

$\dotplus$	<code>\dotplus</code>	$\smallsetminus$	<code>\smallsetminus</code>	$\Cap$	<code>\Cap</code>
$\Cup$	<code>\Cup</code>	$\barwedge$	<code>\barwedge</code>	$\veebar$	<code>\veebar</code>
$\doublebarwedge$	<code>\doublebarwedge</code>	$\boxminus$	<code>\boxminus</code>	$\boxtimes$	<code>\boxtimes</code>
$\boxdot$	<code>\boxdot</code>	$\boxplus$	<code>\boxplus</code>	$\divideontimes$	<code>\divideontimes</code>
$\ltimes$	<code>\ltimes</code>	$\rtimes$	<code>\rtimes</code>	$\leftthreetimes$	<code>\leftthreetimes</code>
$\rightthreetimes$	<code>\rightthreetimes</code>	$\curlywedge$	<code>\curlywedge</code>	$\curlyvee$	<code>\curlyvee</code>
$\circleddash$	<code>\circleddash</code>	$\circledast$	<code>\circledast</code>	$\circledcirc$	<code>\circledcirc</code>
$\centerdot$	<code>\centerdot</code>	$\intercal$	<code>\intercal</code>		

⌘ 17: AMS binary operators

$\leq$	<code>\leqq</code>	$\leq$	<code>\leqslant</code>	$\leq$	<code>\eqslantless</code>
$\lesssim$	<code>\lesssim</code>	$\approx$	<code>\lessapprox</code>	$\approx$	<code>\approxeq</code>
$\lessdot$	<code>\lessdot</code>	$\lll$	<code>\lll</code>	$\lessgtr$	<code>\lessgtr</code>
$\lesseqgtr$	<code>\lesseqgtr</code>	$\lesseqqgtr$	<code>\lesseqqgtr</code>	$\doteqdot$	<code>\doteqdot</code>
$\risingdotseq$	<code>\risingdotseq</code>	$\fallingdotseq$	<code>\fallingdotseq</code>	$\backsimeq$	<code>\backsimeq</code>
$\backsimeq$	<code>\backsimeq</code>	$\subseteq$	<code>\subseteq</code>	$\Subset$	<code>\Subset</code>
$\sqsubset$	<code>\sqsubset</code>	$\preccurlyeq$	<code>\preccurlyeq</code>	$\curlyeqprec$	<code>\curlyeqprec</code>
$\prec$	<code>\prec</code>	$\precapprox$	<code>\precapprox</code>	$\vartriangleleft$	<code>\vartriangleleft</code>
$\trianglelefteq$	<code>\trianglelefteq</code>	$\dashv$	<code>\dashv</code>	$\Vdash$	<code>\Vdash</code>
$\smallsmile$	<code>\smallsmile</code>	$\smallfrown$	<code>\smallfrown</code>	$\bumpeq$	<code>\bumpeq</code>
$\Bumpeq$	<code>\Bumpeq</code>	$\geqq$	<code>\geqq</code>	$\geqslant$	<code>\geqslant</code>
$\eqslantgtr$	<code>\eqslantgtr</code>	$\gtrsim$	<code>\gtrsim</code>	$\gtrapprox$	<code>\gtrapprox</code>
$\gtrdot$	<code>\gtrdot</code>	$\ggg$	<code>\ggg</code>	$\gtrless$	<code>\gtrless</code>
$\gtreqless$	<code>\gtreqless</code>	$\gtreqqless$	<code>\gtreqqless</code>	$\eqcirc$	<code>\eqcirc</code>
$\circeq$	<code>\circeq</code>	$\triangleq$	<code>\triangleq</code>	$\thicksim$	<code>\thicksim</code>
$\thickapprox$	<code>\thickapprox</code>	$\supseteq$	<code>\supseteq</code>	$\Supset$	<code>\Supset</code>
$\sqsupset$	<code>\sqsupset</code>	$\succcurlyeq$	<code>\succcurlyeq</code>	$\curlyeqsucc$	<code>\curlyeqsucc</code>
$\succ$	<code>\succ</code>	$\succapprox$	<code>\succapprox</code>	$\vartriangleright$	<code>\vartriangleright</code>
$\trianglerighteq$	<code>\trianglerighteq</code>	$\Vdash$	<code>\Vdash</code>	$\shortmid$	<code>\shortmid</code>
$\shortparallel$	<code>\shortparallel</code>	$\between$	<code>\between</code>	$\pitchfork$	<code>\pitchfork</code>
$\varpropto$	<code>\varpropto</code>	$\blacktriangleleft$	<code>\blacktriangleleft</code>	$\therefore$	<code>\therefore</code>
$\backepsilon$	<code>\backepsilon</code>	$\blacktriangleright$	<code>\blacktriangleright</code>	$\because$	<code>\because</code>

Fig. 18: AMS binary relations

$\nless$	<code>\nless</code>	$\nleq$	<code>\nleq</code>	$\nleqslant$	<code>\nleqslant</code>
$\nleqq$	<code>\nleqq</code>	$\lneq$	<code>\lneq</code>	$\lneq$	<code>\lneq</code>
$\lvertneqq$	<code>\lvertneqq</code>	$\lnsim$	<code>\lnsim</code>	$\lnapprox$	<code>\lnapprox</code>
$\nprec$	<code>\nprec</code>	$\npreceq$	<code>\npreceq</code>	$\precnsim$	<code>\precnsim</code>
$\precnapprox$	<code>\precnapprox</code>	$\nsim$	<code>\nsim</code>	$\nshortmid$	<code>\nshortmid</code>
$\nmid$	<code>\nmid</code>	$\nvdash$	<code>\nvdash</code>	$\nvDash$	<code>\nvDash</code>
$\ntriangleleft$	<code>\ntriangleleft</code>	$\ntrianglelefteq$	<code>\ntrianglelefteq</code>	$\nsubseteq$	<code>\nsubseteq</code>
$\subsetneq$	<code>\subsetneq</code>	$\varsubsetneq$	<code>\varsubsetneq</code>	$\subsetneqq$	<code>\subsetneqq</code>
$\varsubsetneqq$	<code>\varsubsetneqq</code>	$\ngtr$	<code>\ngtr</code>	$\ngeq$	<code>\ngeq</code>
$\ngeqslant$	<code>\ngeqslant</code>	$\ngeqq$	<code>\ngeqq</code>	$\gneq$	<code>\gneq</code>
$\gneqq$	<code>\gneqq</code>	$\gvertneqq$	<code>\gvertneqq</code>	$\gnsim$	<code>\gnsim</code>
$\gnapprox$	<code>\gnapprox</code>	$\nsucc$	<code>\nsucc</code>	$\nsucceq$	<code>\nsucceq</code>
$\succnsim$	<code>\succnsim</code>	$\succnapprox$	<code>\succnapprox</code>	$\ncong$	<code>\ncong</code>
$\nshortparallel$	<code>\nshortparallel</code>	$\nparallel$	<code>\nparallel</code>	$\nvDash$	<code>\nvDash</code>
$\nVDash$	<code>\nVDash</code>	$\ntriangleright$	<code>\ntriangleright</code>	$\ntrianglerighteq$	<code>\ntrianglerighteq</code>
$\nsupseteq$	<code>\nsupseteq</code>	$\nsupseteqq$	<code>\nsupseteqq</code>	$\supsetneq$	<code>\supsetneq</code>
$\varsupsetneq$	<code>\varsupsetneq</code>	$\supsetneqq$	<code>\supsetneqq</code>	$\varsupsetneqq$	<code>\varsupsetneqq</code>

Fig. 19: AMS negated binary relations

## B 학습을 위한 엄청난 수학 예제

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

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

$$\prod_{j \geq 0} \left( \sum_{k \geq 0} a_{jk} z^k \right) = \sum_{k \geq 0} z^n \left( \sum_{\substack{k_0, k_1, \dots \geq 0 \\ k_0 + k_1 + \dots = n}} a_0 k_0 a_1 k_1 \dots \right)$$

$$(3) \quad \prod_{j \geq 0} \left( \sum_{k \geq 0} a_{jk} z^k \right) = \sum_{k \geq 0} z^n \left( \sum_{\substack{k_0, k_1, \dots \geq 0 \\ k_0 + k_1 + \dots = n}} a_0 k_0 a_1 k_1 \dots \right)$$

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

$$(4) \quad \pi(n) = \sum_{m=2}^n \left[ \left( \sum_{k=1}^{m-1} \left\lfloor \frac{\lfloor m/k \rfloor}{\lceil m/k \rceil} \right\rfloor \right)^{-1} \right]$$

$$\underbrace{\overbrace{a, \dots, a}^{k \text{ a's}}, \overbrace{b, \dots, b}^{l \text{ b's}}}_{k+1 \text{ elements}}$$

$$(5) \quad \underbrace{\overbrace{a, \dots, a}^{k \text{ a's}}, \overbrace{b, \dots, b}^{l \text{ b's}}}_{k+1 \text{ elements}}$$

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

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



## Bibliography

- 1 Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X—A Document Preparation System—User’s Guide and Reference Manual*. Addison-Wesley, Reading, MA, USA, 1985.
- 2 Donald E. Knuth. *The T<sub>E</sub>Xbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986.
- 3 Michel Goossens, Frank Mittelbach and Alexander Samarin. *The L<sup>A</sup>T<sub>E</sub>X Companion*. Addison-Wesley, Reading, MA, USA, 1993.