Essential LaTeX++ 한글판

January 1994 — 2005년 6월 29일(한글판)

Jon Warbrick (원저)
David Carlisle, Michel Goossens, Sebastian Rahtz,
Adrian Clark (추가)
김 재 우 · 김 강 수 (옮김)

---

1 개 요

본 문서는 LaTeX[1] 문서준비시스템을 이용하기 위해서 알아야 할 내용을 알려주고자 한다. 기본적인 내용만 언급하였으며, 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 책자의 내용만 가지고 복잡한 문서를 작성한다면, 보더 더 많은 노력을 들이고도 만족스럽지 못한 결과를 얻게 될 것이다.

가장 널리 이용되는 참고도서는 Leslie Lamport가 쓴 *The LaTeX User's Guide and Reference Manual* 이다. 이 책은 LaTeX 사용자들이 알아야 할 모든 내용을 담고 있으며, LaTeX을 사용하기 위해서는 한 권쯤 갖춰야 할 것이다[2].

매뉴얼이나 이 문서에서는 사용자의 특정 컴퓨터 시스템과 관련된 문서는 언급하지 않는다. LaTeX은 많은 컴퓨터 시스템에서 사용할 수 있고, 그것들은 한두 가지씩 모두 다르기 때문이다. 특정 시스템에 대한 것은 *local guide*[3]를 참조하라.

2 LaTeX은 어떻게 움직이는가?

LaTeX을 사용하기 위해서는, 사용자는 조판할 본문과 본문을 사용자가 원하는 모양이 어떤 것인지를 LaTeX에게 알려주는 지시문을 모두 담고 있는 파일을 작성하여야 한다. 이 파일은 보통 시스템 에디터를 이용하여 작성한다. 파일의 이름은 필요한 대로 정하되 내용물을 식별할 수 있도록 "`.tex`"로 끝나야 한다. 이제 파일을 LaTeX으로 처리하면, 같은 파일이름에 확장자가 "`.tex`"에서 "`.dvi`"로 바뀐 새로운 파일이 생겨난다. 이것은 '*D*evice *I*ndependent'의 약자로서, 이름이 의미하는 바대로 모든 출력장치에 대한 출력물을 작성할 수 있다. 자세한 내용은 *local guide*를 참조하라.

---

[1] 보다 정확히 말해서 1994년 1월 또는 그 이후에 발표된 LaTeX 2_ε를 말한다.

[2] 33페이지를 보라

[3] *local guide*는 각 시스템에 마련되어 있는 TeX, LaTeX 운영에 관한 지침이다. 개인용 컴퓨터에서 LaTeX을 사용하는 경우에는 일반적인 설치 운영 지침을 참고하도록 한다.

LaTeX에서 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. 인간인 조판공은 인용문을 식별하고 그에 따라 취급하지만, LaTeX은 단순히 프로그램에 불과하므로 사용자의 도움을 필요로 한다. 따라서 사용자가 인용문임을 선언하고 LaTeX이 그에 따라 인용문을 정확히 조판하도록 해주는 LaTeX 명령어가 존재하는 것이다.

LaTeX의 기초는 문서가 정확히 어떻게 조판될 것인지를 결정하는 *document class* 개념이다. LaTeX은 인용문과 같은 공통적인 논리 구조를 어떻게 조판할 것인지를 기술한 표준 document class를 제공한다. 사용자는 수학 공식과 같이 사용자 문서에 특정된 논리 구조 형식을 지정함으로써 이러한 스타일을 보충할 수 있다. 비록 완전히 새로운 스타일을 작성하기에 앞서 조판에 대한 기본적인 원칙들을 알고 있어야 하겠지만, 표준 document class를 수정하거나 심지어는 완전히 새롭게 작성하는 것도 가능하다.

문서의 외형이 아니라 논리적 구조에 집중해야 할 이유는 많이 있다. 그렇게 함으로써 문서의 심미적인 질을 높인다는 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 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 작성할 수 있지만 논리적 디자인은 글쓰기에 좀더 집중할 수 있도록 해주어, 형태에 관심을 갖느라고 내용을 희생하는 일을 더 어렵게 한다.

3 샘플 LaTeX 파일

그림 1을 살펴보자.[4] 이것은 표준 LaTeX 예제 파일 `small.tex`을 약간 변형한 것이다. 이러한 파일의 복사본 작성에 관해서는 *local guide*를 참고하라. 좌측에 있는 행번호는 파일의 일부는 아니지만, 참조를 손쉽게 하기 위하여 첨가하였다. 이 파일의 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, 이 점에 있어서는 예제 파일도 예외가 아니다. Running text를 기술하는 데는 아무런 문제가 없다. 일반적인 방법 대로 입력하면 된다. 그에 따라 작성되는 출력물에서, LaTeX은 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

[4] 이 예제 자체가 LaTeX 사용안내이기 때문에 우리말로 옮겨두었다. 34페이지를 보라.

```
 1: % SMALL.TEX -- Released 5 July 1985
 2: % USE THIS FILE AS A MODEL FOR MAKING YOUR OWN LaTeX INPUT FILE.
 3: % EVERYTHING TO THE RIGHT OF A  %  IS A REMARK TO YOU AND IS IGNORED
 4: % BY LaTeX.
 5: %
 6: % WARNING!  DO NOT TYPE ANY OF THE FOLLOWING 10 CHARACTERS EXCEPT AS
 7: % DIRECTED:        &   $   #   %   _   {   }   ^   ~   \
 8:
 9: \documentclass[11pt,a4paper]{article}  % YOUR INPUT FILE MUST CONTAIN THESE
10: \begin{document}                       % TWO LINES PLUS THE \end COMMAND AT
11:                                        % 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 LaTeX 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 파일의 처리 결과

들어 있는 공백이 몇 개이든지간에 LaTeX은 이를 하나의 공백으로 처리한다. 또한 LaTeX은 각 행의 끝을 단어 간의 공백으로 간주한다(15–17행 참조). 입력파일에서 공백인 행은 새로운 단락의 시작임을 가리키므로, 새로운 단락을 시작하는 것이 아니라면 입력파일에 공백인 행을 두지 말아야 한다.

LaTeX은 잘 쓰이지 않는 키보드 문자를 자신의 사용을 위해 확보하고 있다. 사용자의 입력 파일에 이와 같은 열 개의 문자

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

를 사용해서는 안된다—만약 이를 사용한다면, LaTeX은 이를 제대로 처리하지 못하게 된다.

3.2 LaTeX 명령어

입력파일에는 ‘\’로 시작되는 단어들이 있다(9, 10 및 13행 참조). 이들은 LaTeX 명령어로서 사용자 문서의 구조를 기술하는 데 사용된다. 이러한 명령어의 특징은 다음과 같다.

- 모든 LaTeX 명령어는 ‘\’로 시작하고 하나 또는 그 이상의 문자로 이루어진다.
- LaTeX 명령어는 대소문자를 구분한다. 따라서 `\BEGIN`과 `\Begin` 그리고 `\begin`은 각각 서로 다르다.

- 몇몇 명령어들은 사용자의 본문 중에 위치하기도 한다. 이러한 명령어들은 다른 글꼴과 같은 것을 선택하는 스위치와 같은 역할을 한다. `small.tex`에서는 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 활성화하고 '`}`'는 이를 비활성화한다. 이와 같은 글꼴 변경 명령어는 모두 매개변수를 취하는 명령어형을 별도로 가지고 있다. 자세한 것은 9페이지를 참조하라.

- 다음과 같은 형태의 명령어도 있다.

  ```
  \command{text}
  ```

  이 때, `text`를 명령어의 매개변수라고 한다. 명령어 `\section`이 이와 같은 것이다(13행 참조). 매개변수는 중괄호 '`{}`'나 각괄호 '`[]`' 속에 표기한다. 경우에 따라서 두 가지 모두를 사용하기도 한다. 명령어는 있는 그대로 정확하게 사용하여야 한다.

- 명령어가 문자로만 이루어져 있을 때에는 문자가 아닌 것이 명령어의 끝에 오도록 해야 한다. 명령어의 매개변수를 담고 있는 중괄호나 공백을 사용하는 것이 일반적이다. 공백을 사용한 경우에 LaTeX은 그 공백을 무시한다. 이로 인하여 발생할 수 있는 문제에 대해서는 뒤에서 살펴보겠다.

3.3 전체적인 구조

모든 문서에 반드시 사용해야 하는 LaTeX 명령어가 있다. 실제 본문은 `\begin{document}` 명령어로 시작하고 `\end{document}` 명령어로 끝난다(10 및 39행 참조). `\end{document}` 이후에 있는 모든 것은 무시된다. `\begin{document}` 이전에 나타나는 것들을 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 지정한다. 예제 파일은 간단한 기술문서이므로 article 형식을 사용하고, 11포인트 글꼴로 인쇄하도록 수정하였다. 제 4 절에 나타나 있는 것과 같은 다른 형식의 사용도 가능하다.

3.4 기타 살펴볼 사항

LaTeX은 열고 닫는 따옴표와 쌍따옴표를 모두 식자할 수 있다. 이는 키보드에 있는 두 개의 따옴표—'(grave accent 또는 back-quote와 유사하다)와 '(apostrophe)—를 이용한 것이다. 따옴표에는 한 번(21행 참조), 쌍따옴표에는 두 번을 사용한다(20행 참조). 쌍따옴표 문자 자체 "는 거의 사용하지 않는다.

LaTeX은 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 LaTeX에서 사용하는 특수 문자를 입력해야 할 필요가 있다. 그 중 일곱 개의 문자는 앞에서 역슬래시를 붙여 입력할 수 있다(36행 참조). 나머지 세 개의 문자는 †, ‡, §, £, ©, ♯, ♣와 같이 키보드에 없는 문자를 입력하는 것과 마찬가지로 별도의 명령어를 이용해 작성한다.

자신이 작업한 내용이나 이유 등을 알 수 있도록 LaTeX 파일에 주석을 첨가해 둔다면 매우 유용할 것이다. LaTeX은 % 기호 우측에 있는 모든 것을 무시하므로 이를 주석문으로 사용한다.

4 도큐먼트 클래스와 옵션

LaTeX에는 네 종류의 표준 도큐먼트 클래스가 있다.

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

`report` 좀더 기술적인 문서 작성에 알맞다. 이는 chapter를 가지며 제목을 별도의 페이지에 인쇄하는 점을 제외하면 article과 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.

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 컴마로 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` article이나 report 형식에서 문서를 종이의 양면에 인쇄하도록 한다. 이는 book 형식에서는 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 a4paper는 모든 표준 형식의 출력을 A4 용지에 맞게끔 조정한다. (LaTeX이 만들어진 미국에서는 A4 용지보다 길이가 조금 짧고 폭이 넓은 용지가 표준이어서, 이 옵션 없이 인쇄하면 어딘지 모르게 어색할 것이다.)

5 환경

우리는 앞에서 LaTeX이 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 LaTeX 구조의 한 예이다. 본문을 특정 환경에 배치하면 여러 가지 특수한 효과를 얻을 수 있다.

5.1 인용

인용에는 quote와 quotation의 두 가지 환경이 있다. quote는 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.

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}
```

5.2 중앙 및 좌우정렬

center 환경에 배치된 본문은 페이지의 중앙에 정렬되고, flushleft나 flushright 환경에 배치된 본문은 좌·우측 끝에 정렬된다. center의 철자에 주의하라—불행히도 LATEX은 영국식 철자(centre)를 받아들이지 못한다.

이와 같은 환경 속에 있는 본문은 일반적인 방식 대로 조판된다. 특히 사용자가 입력한 행의 끝 문자는 단순히 공백으로 간주된다. 새로운 행임을 알리기 위해서는 \\ 명령어를 입력하여야 한다. 예를 들어보자.

one two three
four
five

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

5.3 리스트

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}
```

리스트를 작성하기 위한 세번째 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 원형 출력

경우에 따라서는 터미널 화면에 나타난 모양 대로 본문을 입력할 필요가 있다. 컴퓨터 프로그램의 입력과 같은 것이 그 예가 될 것이다. 사용자가 원하는 것은 LaTeX이 사용자의 본문 배치를 중단하는 것만이 아니라 LaTeX에게 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}
```

6 글꼴

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

Command	*or*	*Effect*
`\textrm{...}`	`{\rmfamily...}`	Text is set in roman family
`\textsf{...}`	`{\sffamily...}`	Text is set in sans serif family
`\texttt{...}`	`{\ttfamily...}`	Text is set in typewriter family
`\textmd{...}`	`{\mdseries...}`	Text is set in medium series
`\textbf{...}`	`{\bfseries...}`	Text is set in **bold** series
`\textup{...}`	`{\upshape...}`	Text is set in upright shape
`\textit{...}`	`{\itshape...}`	Text is set in *italic* shape
`\textsl{...}`	`{\slshape...}`	Text is set in *slanted* shape
`\textsc{...}`	`{\scshape...}`	Text is set in SMALL CAPS shape
`\emph{...}`	`{\em ...}`	Text is set *emphasized*
`\textnormal{...}`	`{\normalfont...}`	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 본문을 매개변수로 하는 명령형(첫번째 열)의 사용을 권한다. 글꼴 명령에 더하여, 글꼴의 크기를 변경하는 명령어도 있다.

tiny scriptsize footnotesize small normal-

size 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

이 문서와 같은 기술 관계 문서들은 보통 여러 섹션으로 구분된다. 각각의 섹션은 참조를 보다 손쉽게 하기 위해서 제목과 번호를 부여한다. LaTeX에는 각기 다른 종류의 섹션을 식별하기 위한 일련의 명령어가 준비되어 있다. 일단 사용자가 이러한 명령어를 사용하면, 제목의 배치와 번호의 부여는 LaTeX이 처리한다.

이러한 명령어에는 다음과 같은 것들이 있다.

```
\chapter        \subsection             \paragraph
\section        \subsubsection          \subparagraph
```

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 chpater 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. 이 명령어는 다른 명령어에서 부여한 번호를 그대로 보존한다.

사용자는 문서에서 `\tableofcontents`를 사용하여 각종 섹션 명령을 바탕으로 하여 작성된 내용에 따른 목차를 작성·포함할 수 있다. 문서를 매번 LaTeX으로 처리할 때마다 직전에 LaTeX 처리에서 얻어진 제목 등으로 목차가 새롭게 구성된다는 점에 유의하여야 한다. 이는 LaTeX 문서를 처리할 때 목차에 필요한 정보를 모아두었다가 다시 문서를 처리할 때 이 정보를 이용하기 때문이다. 이 때문에 정확한 목차를 작성하기 위해서 문서를 LaTeX으로 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}   ǒ \v{o}   ǫ \c{o}   ó \’{o}
ō \={o}   ő \H{o}   ọ \d{o}   ô \^{o}   ȯ \.{o}
o͡o \t{oo}   o̲ \b{o}
ö \"{o}   ŏ \u{o}
```

기타의 특수기호와 그 입력 명령어를 아래에 표시하였다.

†	\dag	§	\S	©	\copyright
‡	\ddag	¶	\P	£	\pounds
œ	\oe	Œ	\OE	æ	\AE
Æ	\AE	å	\aa	Å	\AA
ø	\o	Ø	\O	ł	\l
Ł	\E	ß	\ss	¿	?‘
¡	!‘	...	\ldots	LaTeX	\LaTeX

명령어 `\today`는 현재 날짜를 문서에 삽입한다. 이러한 LaTeX 명령어를 사용할 때에는 LaTeX이 명령어 뒤의 모든 공백을 무시한다는 점에 유의해야 한다. 따라서 ‘`\pounds 20`’라고 입력하면 ‘£20’과 같이 나타난다. 어쨌든 ‘`\LaTeX is wonderful`’라고 입력하면 ‘LaTeXis wonderful’이라고 나타나는데, LaTeX 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 ‘LaTeX really is wonderful!’과 같이 나타날 것이다.

마지막으로 보통 수식의 조판에 사용하는 LaTeX의 수식 모드에서는, 대·소문자의 희랍어 수학 알파벳과 멋진 글꼴 그리고 수학 연산자와 관계 연산자 및 화살표와 기타 많은 것을 포함하여 훨씬 더 많은 기호를 사용할 수 있다. 이에 관해서는 제 13 절에서 자세히 다룬다.

9 제목

대부분의 문서에는 제목이 있다. LaTeX 문서에 제목을 주기 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 \and로 구분하여 표시하면 각자의 이름이 서로 다른 행에 중앙정렬되어 인쇄된다. \date 명령을 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
```

제목의 정확한 모양은 도큐먼트 형식에 따른다. report와 book 형식에서는 표지 페이지에 인쇄되고, article 형식에서는 첫번째 페이지의 상단에 인쇄된다. 클래스 옵션인 titlepage를 이용하면 이를 변경할 수 있다(제 4 절 참조).

10 표의 사용

LATEX은 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. \kill로 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 LATEX이 전체 표를 검토하여 가장 긴 항목을 집어넣기 위해서는 열의 간격을 어떻게 해야 할지를 결정하기 때문에 tabbing에서와 같이 열의 너비에 대해 염려하지 않아도 된다. 사용자는 단지 몇 개의 열이 있고 각 열을 어떻게 배치하고 싶은지를 LATEX에게 알려주기만 하면 된다. 이는 \begin{tabular} 명령 다음에 각 열에 대해 한 문자로 된 *template*을 이용하여 지정한다.

`l` means the column will be left justified
`r` means the column will be right justified
`c` means the column will be centered

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.

Name	*Age*	*Height*
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 `|`를 사용하면 열과 열 사이에 선을 그을 수 있다.
3. *template*에 `p`를 사용하고 `{`와 `}` 사이에 폭을 지정함으로써 열의 폭을 고정시킬 수 있다. 이 때 폭은 'cm', 'mm' 또는 'in'으로 표기할 수 있다('inches'를 사용할 수는 없다).

좀더 세밀한 표를 작성해보자.

Group	*Type*	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 miscellaneous	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 \\
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 \\ \hline
  && 538 \\ \hline
\end{tabular}
```

사용자가 입력한 행의 끝이 중요한 게 아니라는 데 주목하라. 행의 끝은 반드시 \\로 이루어진다.

11 표와 그림

표에 문자와 숫자를 배치함에 있어서, 서로 다른 페이지에 나뉘어 배치하는 게 아니라 거의 항상 전체 표를 함께 배치하여야 할 것이다. 페이지의 아랫부분에서 충분한 공간이 확보되지 못할 경우에는 어떻게 할 것인가? 이와 비슷한 문제로서, 나중에 사진을 붙여넣기 위한 공간을 어떻게

확보할 것인가? 사용자가 본문을 입력할 당시에는 LATEX이 어디에서 새로운 페이지를 시작할지 알 수 없으므로 필요한 만큼의 공간을 확보하기가 매우 어렵다. LATEX은 정확히 사용자의 입력 파일에 있는 그 위치가 아니라 페이지 내의 적당한 위치에다 이를 배치하는 이동개체(*float object*) 시스템을 이용하여 이 문제를 해결하고 있다. 보통 현재 페이지의 하단이나 다음 페이지의 상단 또는 더 많은 공간이 필요하다면 개체만의 페이지에 배치한다. LATEX은 두 종류의 이동개체를 제원하는데, *table*과 *figure*라고 하는 것이 그것으로서, 이들은 각기 캡션을 지정할 수 있다. 그림을 표시하기 위해 3인치의 공간을 확보하고 캡션을 지정하는 예는 다음과 같다.

```
\begin{figure}
\vspace{3in}
\caption{A Photopraph 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 목록을 캡션과 함께 표시해준다.

12 상호참조와 인용

LATEX의 가장 유용한 기능 중의 하나가 상호참조에 사용될 수식, 표, 그림, 페이지, 섹션 등의 번호를 자동으로 생성해주는 것일 것이다. 예를 들어, 도큐먼트 스타일에 대해서는 제 4 절을 참조하고, `sample.tex`의 처리 결과는 그림 2를 참조하는 것과 같은 것은,

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

와 같이 표시된다. 문서에서 참조번호가 인쇄된 곳에는 번호 대신에 `\ref` 명령어가 있음을 알 수 있다. `~`는 LATEX이 줄바꿈을 할 수 없는 공백을 삽입한다.

상호참조를 하기 위해서는 LATEX 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}
```

\label과 \ref에는 어떠한 이름일지라도 사용할 수는 있지만, 위의 관례를 따르는 것도 나쁘지 않다. \label은 페이지 참조에도 사용된다. 다만 \ref가 아니라 \pageref를 사용한다.

LaTeX 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 매개변수는 *bibliographic database*에 들어 있는 책자나 논문을 구분·인식하는 데 이용된다.

문서에서 다른 문서를 인용한다면, 다음의 명령어로 참조목록을 만들 수 있다.

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

LaTeX에서 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 \bibliography의 매개변수는—위의 예제에서 `mybib1.bib`나 `mybib2.bib`와 같이—확장자가 `.bib`인 파일 이름을 컴마로 구분하여 표기한다. 이 파일에는 \cite 명령에 대한 완전한 참조가 들어 있다. 이러한 `.bib` 파일의 형식이나, 인용을 어떻게 LaTeX과 호환성있는 형식으로 변환하느냐 하는 등의 문제는 이 안내서의 주제를 넘어서는 것이다. 참고문헌에 있는 Lamport의 책([1])에 자세히 설명되어 있다.

13 수식의 조판

13.1 Math, Display-math 그리고 Equation

TeX은 수식을 일반적인 텍스트와는 완전히 달리 취급한다. 수식을 위해서는 *math mode*와 *display math mode*라고 알려진 두 가지의 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 mode는 `\[...\]`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 equation은 자동적으로 수식에 번호를 붙인다.

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

```
\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 $\infty \Rightarrow \surd \bigotimes$와 같이 표시된다.
4. 배열은 `array` 환경으로 표현한다. 이는 각 항목이 LR 모드가 아니라 math mode로 조판된다는 점을 제외하면 제 10 절에서 설명한 `tabular` 환경과 동일하다. `array` 환경에서는 배열에 괄호를 표시하지 않는 데 주목하여야 한다. 따라서 행렬식이나 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 8과 표 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

LATEX에서 간격 조정이 필요한 좋은 예가 있다.

$$\iint z\,dxdy..\int\int zdxdy$$

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

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.

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

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

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

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

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

$\mathcal{F}$

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

13.4 $가 의미하는 것은?

$\mathcal{AMS}$-TEX이나 plain TEX을 사용하다가 LATEX으로 전환했다면, 왜 $나 $$에 대해 언급이 없는지 궁금할 것이다.

이 시스템들은 math mode를 $로 둘러싸서 표시하고, display math mode는 $$로 둘러싸서 표시한다. 이는 문서 편집기에서 $ 기호의 짝을 맞추기가 어렵고 math mode의 시작과 끝이 어딘지를 알기가 어려우므로 LATEX 시스템에 비해 단점이 된다. 사용자가 $ 기호를 빠뜨리면 TEX도 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 메세지를 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. TEX은 'be a ... such that'을 수식 모드로 조판하고 'such that' 다음의 $에서 수식 모드를 마친다.

동일한 실수를 LATEX에서 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 메세지의 내용은 여전히 멍청하지만, (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 조그만 수식의 표현에 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 1에서 표 19까지 LaTeX의 표준 기호를 종류별로 정리해두었다. 표 12에서 표 18에 있는 $\mathcal{AMS}$-TeX의 추가적인 기호들은 amssymb 패키지를 지정해야지만 사용할 수 있다.

이 표들은 표준 LaTeX 기호 글꼴에서 사용가능한 기호 대부분을 보여준다. 관계 연산자에 대한 부정은 \not로 이루어진다.

$G \not\equiv H$

```
$G \not\equiv H$
```

14 편지

LaTeX으로는 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 매개변수로 하는 letter 환경으로 작성된다. 편지의 본문은 \opening 명령에 따라 인사와 함께 시작된다.

편지는 \closing 명령으로 끝난다. 첨부물이나 사본의 배부처를 표시하기 위해서는 \encl이나 \cc 명령어를 사용한다. 명령어 \closing 뒤의 모든 내용은 \ps를 표기한 후에 기록하여야 한다. 이 명령어는 아무런 표기도 하지 않으므로 사용자는 "추신"이라고 직접 표시하여야 하지만, 추가되는 문장을 정확하게 배치하기 위해서 필요한 명령이다.

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. Natheniel 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 LaTeX 입력 파일을 작성하다보면 아마 실수를 하게 될 것이다. 누구나가 그런 실수는 하는 것이므로 그렇게 염려할 필요는 없다. 여기에는 다른 컴퓨터 프로그램과 같이 두 종류의 실수가 있다. LaTeX에서 경고를 해주는 것과 그렇지 않은 것 두 가지 종류이다. 간단히 예를 들자면, 사용자의 본문을 잘못 입력했다면 LaTeX은 이를 알지 못하고 사용자는 결과물을 알아서 읽으면 된다. 반면에 LaTeX의 환경명을 잘못 입력했다면 LaTeX은 어떻게 해야 할지를 알 수 없게 된다.

이러한 상황이 발생하면 LaTeX은 화면에 에러 메세지를 출력한 뒤 처리를 중단하고 사용자의 조치를 기다린다. 생성된 에러 메세지는 사용자가 이해하기 쉬울 것으로 생각한다. 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 LaTeX은 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
l.140 \begin{itemie}

?
```

LaTeX은 '?'를 출력한 뒤 사용자가 어떤 지시를 하는지 기다리게 된다.

처음의 두 줄은 단지 LaTeX이 에러를 감지하였음을 알려주는 것뿐이다. '!'로 시작하는 세번째 줄은 에러 지시문이다. LaTeX에 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 LaTeX이 무엇을 하고 있었는지를 알려준다. 지금은 별 관계가 없는 내용이므로 무시한다. 마지막 줄은 에러 위치를 나타내며, 사용자의 입력 파일에서 문제가 된 줄을 복사하여 나타낸다. 사용자가 손쉽게 찾을 수 있도록 행번호로 시작한다. 에러가 라인 중간에서 발생하였다면, LaTeX이 에러라고 인식한 그 부분에서 행을 분리하여 표시한다. 다른 모든 컴퓨터 프로그램과 마찬가지로 LaTeX에서도 가끔씩 잘못을 발견하기 전에 실제 에러가 있는 부분을 지나쳐 버리기도 한다. 하지만 대부분 그리 멀지는 않다.

이 경우 몇 가지 행동을 취할 수 있다. LaTeX을 잘 알고 있다면 문제를 수정할 수도 있을 것이고, 그렇지 않다면 'x'를 입력하여 LaTeX의 가동을 중단시킨 다음 에러를 수정할 수도 있을 것이다. 제일 좋은 방법은 그냥 리턴키를 누르는 것이다. 그러면 LaTeX은 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 LaTeX을 실행해서 하나씩 에러를 수정하는 것보다 훨씬 효율적일 것이다. 나타나는 모든 메세지는 나중의 참고를 위해 *log* 파일에 저장되므로 어떤 에러가 있었는지 기억할 필요는 없다. 이 파일을 무엇이라고 하는지는 각 시스템의 *local guide*를 참고하라. 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 Lamport의 책([1]) 제 6 장을 참고하라. 접하게 되는 에러 메세지 전부를 수록하고 있으며, 에러가 일어난 원인 등에 대한 제안도 함께 담고 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 매개변수(중괄호 사이에 있다)를 빠뜨린 경우.

하나의 에러는 LaTeX을 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 LaTeX은 에러 메세지도 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 LaTeX은 가끔씩 *warning* 메세지를 출력한다. 이것은 LaTeX의 작업을 중단할 것까지는 없으나 조사해볼 필요가 있는 문제가 있음을 보고하는 것이다. 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 LaTeX이 행을 분리할 적절한 위치를 찾지 못했음을 알린다. 그 결과 해당 행은 우측 경계 밖으로 삐어져 나가게 배치해 버린다. 이 경우에는 10.6포인트만큼이다. 1포인트는 1/72.27 인치이므로, 이를 알아보기는 어렵다. 하지만 아무것도 아닌 것은 아니다.

이는 LaTeX이 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 LaTeX에게 알려주는 것이다.

이와는 달리 "Underfull \hboxes" 메세지가 나타날 수도 있다. 이는 단어와 단어 사이에 LaTeX이 원하는 것보다도 더 많은 공백이 있는 행임을 의미한다. 일반적으로 이 경우에 할 수 있는 일이란 별로 없다. 행이 늘여진 느낌이 있겠지만, 출력물은 괜찮아 보일 것이다. 이와 같은 경우에 할 수 있는 유일한 것은 문제의 단락을 재작성하는 것이다!

16 마치는 글

이제 다양한 종류의 문서를 작성할 수 있을 정도로 LaTeX에 대해 많이 알게 되었을 것이다. 하지만 이 문서에서 다룬 것은 LaTeX이 할 수 있는 일에 비한다면 겨우 수박의 겉을 핥은 것에 지나지 않는다. 이 문서 전체를 LaTeX으로 작성하였다(속임수나 복사기를 사용하지 않고서). 게다가 사용할 수 있는 모든 특성을 다 사용한 것도 아니다. 이로 미루어 여러분의 처분에 맡겨져 있는 LaTeX의 파워를 조금이나마 느낄 수 있을 것이다.

복잡한 문서를 작성하고자 한다면 Lamport의 책([1])을 참조하라. 또한, 이 문서의 개요에서 얘기한 내용을 기억하라. 여기에서 얻은 내용에만 머무른다면 시간만 낭비하게 될 것이다. LaTeX을 어느 정도 사용하고 나면, 문서를 조판하는 LaTeX 사용자들을 돕기 위해 개발된 많은 기능 확장 파일들을 어떻게 사용하는지 알고 싶을 것이다. 이에 관한 보다 많은 것들(색인의 작성을 위해 makeindex를 어떻게 사용하는지, 또는 참고문헌을 관리하기 위해 BibTeX을 어떻게 사용하는지와 같은)을 *The LaTeX Companion*[3]에서 다루고 있다. 마지막으로 TeX에 대한 궁극적인 한 마디는 Knuth 교수의 *TeXbook*[2]이다.

그리고 또다른 경고 한 마디, LaTeX을 가지고 놀다보면 여러분의 문서가 결코 다시 예전과 같아질 수는 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] Michel Goossens, Frank Mittelbach and Alexander Samarin, *The LaTeX Companion*. Addison-Wesley, Reading, MA, USA, 1993.

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. 더 완전한 기호의 리스트는 Scott Pakin이 작성한 *The Comprehensive LaTeX Symbols List*[5]를 참고하라.

표 1: 수학 모드의 액센트 기호.

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

표 2: 그리스 문자 소문자.

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		

표 3: 그리스 문자 대문자.

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

[5] http://ftp.ktug.or.kr/mirrors/CTAN/info/symbols/comprehensive/symbols-letter.pdf

표 4: Binary Relations.

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

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

[a] Use the latexsym package to access this symbol

표 5: Binary Operators.

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

표 6: BIG Operators.

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

표 7: 화살표.

$\leftarrow$	\leftarrow or \gets	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\rightarrow$	\rightarrow or \to	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons	$\Longleftrightarrow$	\iff (bigger spaces)	$\rightsquigarrow$	\leadsto [a]

[a] Use the latexsym package to access this symbol

표 8: 짝맞춤 문자(Delimiters).

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

표 9: 큰 짝맞춤 문자(Large Delimiters).

$\left($	\lgroup	$\right)$	\rgroup	$\left\int$	\lmoustache	$\right\rceil$	\rmoustache
$\left\vert$	\arrowvert	$\left\Vert$	\Arrowvert	$\left\vert$	\bracevert		

표 10: 그밖의 기호.

…	\dots	⋯	\cdots	⋮	\vdots	⋱	\ddots
ℏ	\hbar	ı	\imath	ȷ	\jmath	ℓ	\ell
ℜ	\Re	ℑ	\Im	ℵ	\aleph	℘	\wp
∀	\forall	∃	\exists	℧	\mho [a]	∂	\partial
′	’	′	\prime	∅	\emptyset	∞	\infty
∇	\nabla	△	\triangle	□	\Box [a]	◊	\Diamond [a]
⊥	\bot	⊤	\top	∠	\angle	√	\surd
♢	\diamondsuit	♡	\heartsuit	♣	\clubsuit	♠	\spadesuit
¬	\neg or \lnot	♭	\flat	♮	\natural	♯	\sharp

[a] Use the latexsym package to access this symbol

표 11: Non-Mathematical Symbols.

다음 기호들은 텍스트 모드에서도 사용할 수 있다.

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

표 12: AMS Delimiters.

⌜	\ulcorner	⌝	\urcorner	⌞	\llcorner	⌟	\lrcorner
|	\lvert	|	\rvert	‖	\lVert	‖	\rVert

표 13: AMS Greek and Hebrew.

ϝ	\digamma	ϰ	\varkappa	ב	\beth	ג	\gimel	ד	\daleth

표 14: AMS Binary Relations.

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

표 15: AMS Arrows.

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

표 16: AMS Negated Binary Relations and Arrows.

$\not<$	\nless	$\not>$	\ngtr	$\subsetneqq$	\varsubsetneqq
$\lneq$	\lneq	$\gneq$	\gneq	$\supsetneqq$	\varsupsetneqq
$\not\leq$	\nleq	$\not\geq$	\ngeq	$\not\subseteqq$	\nsubseteqq
$\not\leqslant$	\nleqslant	$\not\geqslant$	\ngeqslant	$\not\supseteqq$	\nsupseteqq
$\lneqq$	\lneqq	$\gneqq$	\gneqq	$\nmid$	\nmid
$\lvertneqq$	\lvertneqq	$\gvertneqq$	\gvertneqq	$\nparallel$	\nparallel
$\not\leqq$	\nleqq	$\not\geqq$	\ngeqq	$\nshortmid$	\nshortmid
$\lnsim$	\lnsim	$\gnsim$	\gnsim	$\nshortparallel$	\nshortparallel
$\lnapprox$	\lnapprox	$\gnapprox$	\gnapprox	$\not\sim$	\nsim
$\not\prec$	\nprec	$\not\succ$	\nsucc	$\not\cong$	\ncong
$\not\preceq$	\npreceq	$\not\succeq$	\nsucceq	$\nvdash$	\nvdash
$\precneqq$	\precneqq	$\succneqq$	\succneqq	$\nvDash$	\nvDash
$\precnsim$	\precnsim	$\succnsim$	\succnsim	$\nVdash$	\nVdash
$\precnapprox$	\precnapprox	$\succnapprox$	\succnapprox	$\nVDash$	\nVDash
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq	$\not\triangleleft$	\ntriangleleft
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq	$\not\triangleright$	\ntriangleright
$\not\subseteq$	\nsubseteq	$\not\supseteq$	\nsupseteq	$\not\trianglelefteq$	\ntrianglelefteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq	$\not\trianglerighteq$	\ntrianglerighteq
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow	$\nLeftrightarrow$	\nLeftrightarrow

표 17: AMS Binary Operators.

$\dotplus$	\dotplus	$\centerdot$	\centerdot	$\intercal$	\intercal
$\ltimes$	\ltimes	$\rtimes$	\rtimes	$\divideontimes$	\divideontimes
$\Cup$	\Cup or \doublecup	$\Cap$	\Cap or \doublecap	$\smallsetminus$	\smallsetminus
$\veebar$	\veebar	$\barwedge$	\barwedge	$\doublebarwedge$	\doublebarwedge
$\boxplus$	\boxplus	$\boxminus$	\boxminus	$\circleddash$	\circleddash
$\boxtimes$	\boxtimes	$\boxdot$	\boxdot	$\circledcirc$	\circledcirc
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes	$\circledast$	\circledast
$\curlyvee$	\curlyvee	$\curlywedge$	\curlywedge		

표 18: AMS Miscellaneous.

$\hbar$	\hbar	$\hslash$	\hslash	$\Bbbk$	\Bbbk
$\square$	\square	$\blacksquare$	\blacksquare	Ⓢ	\circledS
$\vartriangle$	\vartriangle	$\blacktriangle$	\blacktriangle	$\complement$	\complement
$\triangledown$	\triangledown	$\blacktriangledown$	\blacktriangledown	$\Game$	\Game
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge	$\bigstar$	\bigstar
$\angle$	\angle	$\measuredangle$	\measuredangle	$\sphericalangle$	\sphericalangle
$\diagup$	\diagup	$\diagdown$	\diagdown	$\backprime$	\backprime
$\nexists$	\nexists	$\Finv$	\Finv	$\varnothing$	\varnothing
$\eth$	\eth	$\mho$	\mho		

표 19: Math Alphabets.

Example	Command	Required package
ABCdef	\mathrm{ABCdef}	
$ABCdef$	\mathit{ABCdef}	
$ABCdef$	\mathnormal{ABCdef}	
$\mathcal{ABC}$	\mathcal{ABC}	
$\mathfrak{ABCdef}$	\mathfrak{ABCdef}	
$\mathbb{ABC}$	\mathbb{ABC}	amsfonts or amssymb

B 학습을 위한 엄청난 수학 예제

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

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

$$\prod_{j\geq 0} \left(\sum_{k\geq 0} a_{jk} z^k \right) = \sum_{k\geq 0} z^n \left(\sum_{\substack{k_0,k_1,\ldots\geq 0 \\ k_0+k_1+\ldots=n}} a_0 k_0 a_{1k_1} \ldots \right) \tag{2}$$

```
\begin{equation}
  \prod_{j\geq 0}
  \left(\sum_{k\geq 0}a_{jk} z^k\right)
= \sum_{k\geq 0} z^n
  \left( \sum_{{k_0,k_1,\ldots\geq 0}
           \atop{k_0+k_1+\ldots=n} }
       a{_0k_0}a_{1k_1}\ldots \right)
\end{equation}
```

$$\pi(n) = \sum_{m=2}^{n} \left\lfloor \left(\sum_{k=1}^{m-1} \lfloor (m/k)/\lceil m/k \rceil \rfloor \right)^{-1} \right\rfloor \tag{3}$$

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

$$\{\overbrace{\underbrace{a,\ldots,a}_{k\ a's}, \underbrace{b,\ldots,b}_{l\ b's}}^{k+1\ \text{elements}}\} \tag{4}$$

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

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

```
\begin{displaymath}
\mbox{W}^+\
\begin{array}{l}
\nearrow\raise5pt\hbox{$\mu^+
+ \nu_{\mu}$}\\
\rightarrow \pi^+ +\pi^0 \\[5pt]
\rightarrow \kappa^+ +\pi^0 \\
\searrow\lower5pt\hbox{$\mathrm{e}^+
       +\nu_{\scriptstyle\mathrm{e}}$}
\end{array}
\end{displaymath}
```

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

```
\begin{displaymath}
{F}(x,y)=0\quad\mathrm{and}\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}
```

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

```
\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}
```

$$\begin{aligned}\sigma_0^f(Q,T_{3R},\beta,s) &= \frac{4\pi\alpha^2}{3s}\beta\times\left[\frac{Q^2\left\{\frac{3-\beta^2}{2}\right\}-2QC_{\rm V}C'_{\rm V}s(s-M_{\rm Z}^2)}{(s-M_{\rm Z}^2)^2+M_{\rm Z}^2\Gamma_{\rm Z}^2\left\{\frac{3-\beta^2}{2}\right\}}\right. \\ &+ \left.\frac{(C_{\rm V}^2+C_{\rm A}^2)s^2}{(s-M_{\rm Z}^2)^2+M_{\rm Z}^2\Gamma_{\rm Z}^2\left\{C'^2_{\rm V}\left\{\frac{3-\beta^2}{2}\right\}+C'^2_{\rm A}\{\beta^2\}\right\}}\right] \end{aligned} \tag{5}$$

```
\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}{\MZs^2 + \MZ \GZ \BE}
         \right.                                          \nonumber \\[-3mm]
                &           &                                      \\[-3mm]
                &    +      &
   \left.\frac{(\CV^2 + \CA^2) s^2}%
              {\MZs^2+\MZ\GZ\left\{\CPV^2 \BE+\CPA^2 \{\beta^2\}\right\}}
      \right] \nonumber
\end{eqnarray}
```

C 역자의 말

이 문서에 대하여. 이 문서는 J. Warbrick 씨가 작성한 문서(CTAN에서 찾을 수 있는 `essential.tex`)를 토대로 David Carlisle 씨 등이 수식에 관한 내용 등을 보충하여 만든 "Essential LaTeX++"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 조성한 것이다.

주의사항. 이 글은 1994년에 쓰여진 것으로서, 지난 10년간의 LaTeX 발전을 반영하고 있지 않다. 사용자는 이 글의 내용을 대부분 신뢰해도 좋지만, 보다 쉬운 해결책과 다양한 스타일들이 개발되어 쓰이고 있다는 점에도 주의를 기울여야 할 것이다. 이러한 개선들은 대부분 *style packages*의 형태로 CTAN을 통해 제공되고 있으며, 그 사용법은 *The LaTeX Companion*에서 찾아볼 수 있다.

이 글의 내용과 관련된 것은 대강 다음과 같다.

1. Lamport의 책은 그 후 개정되지 않았다. 만약 참고문서가 필요하다면 *The LaTeX Companion*[3]의 최신판(제2판)을 찾아볼 것을 권장한다.
2. LaTeX의 표준 도큐먼트 클래스로 제시된 `article`, `report`, `book`, `letter` 이외에 `memoir` 클래스는 이전의 `book` 클래스보다 훨씬 유연하고 강력한 문서 작성 도구를 제공한다.
3. 원형출력(`verbatim`)과 관련된 많은 개선이 이루어져 있다. 예를 들면, `fancyvrb`, `verbatim`, `sverb` 패키지 등은 이전의 단순히 `verbatim` 환경만으로는 표현할 수 없었던 많은 추가적인 기능을 제공한다.
4. 글꼴은 표준적인 Computer Modern 글꼴 이외에도 Type 1 글꼴을 비롯하여 truetype에 이르기까지 괄목할 만한 개선이 이루어져 있다.
5. 표 작성에 관련한 많은 스타일 패키지들이 있다. 예를 들면 `array` 패키지를 참고하라.

이 문서가 다루고 있지 않은 것은 다음과 같은 사항들이다.

1. 그림에 관련된 내용은 이 문서에서는 거의 다루어지지 않았다.
2. 다국어 문서의 조판에 관한 사항은 다루어지지 않았다.
3. 글꼴 사용에 관한 사항은 불충분하다.
4. PDF 문서 만들기와 관련된 사항이 충분히 언급되고 있지 않다.

이 글을 읽은 후에 LaTeX의 작용 방식에 관해서 어느 정도 이해하였다면, 이보다 조금 더 자세한 참고문서를 읽어볼 것을 권장한다.

D 예제 문서(한글판)

2페이지에 있는 예제 small.tex의 한글 번역이다. 입력방법과 결과를 유심히 대조해보라. 한 가지 다른 점은 한글을 쓰기 위해서 \usepackage{hangul}을 추가한 것이다(10행 참조).

```
 1: % SMALL.TEX -- Released 5 July 1985
 2: % USE THIS FILE AS A MODEL FOR MAKING YOUR OWN LaTeX INPUT FILE.
 3: % EVERYTHING TO THE RIGHT OF A  %  IS A REMARK TO YOU AND IS IGNORED
 4: % BY LaTeX.
 5: %
 6: % WARNING!  DO NOT TYPE ANY OF THE FOLLOWING 10 CHARACTERS EXCEPT AS
 7: % DIRECTED:        &   $   #   %   _   {   }   ^   ~   \
 8:
 9: \documentclass[11pt,a4paper]{article}  % 모든 \LaTeX 문서는 이 문장으로 시작한다.
10: \usepackage{hangul}                    % 한글 사용을 위한 설정
11: \begin{document}                       % 이 행과 문서 제일 마지막의 \end 명령을
12:                                        % 반드시 포함해야 한다.
13: \section{단순한 문장}                  % 이 명령은 섹션 타이틀을 만든다.
14:
15: 단어들은 하나 또는     그     이상의 스페이스로 나누어진다. 단락은
16:     하나 또는 그 이상의 빈 줄로 나누어진다. 출력결과는
17: 입력 파일에 빈 스페이스나 빈 줄을 얼마든지 써넣어도 달라지지 않는다.
18:
19:
20: 겹따옴표는 이렇게 타이프한다: ``따옴표친 문구''.
21: 홑따옴표는 이렇게 타이프한다: `홑따옴표친 문구'.
22:
23: 긴 대시는 세 개의 대시 문자를 써넣는다---이런 식으로.
24:
25: 이탤릭체 텍스트는 이렇게 한다: \emph{이탤릭체 텍스트}.
26: 굵은 글꼴 텍스트는 이렇게 한다: \textbf{굵은 글꼴 텍스트}.
27:
28: \subsection{간단한 경고}          % 이 명령은 서브섹션 타이틀을 만든다.
29:
30: 문장의 끝이 아닌 위치에 찍은 마침표 다음에 나오는 공백이 너무 큰 경우---예를 들면
31: etc.\ 와 같은 일반적인 약자를 쓰는 경우 따위---에는, 마침표 뒤에 역슬래시 하나와
32: 빈 스페이스 하나를 두라. 이 문장을 보면 알 수 있을 것이다.
33:
34: 열 개의 특별한 문자들(예컨대 달러 기호나 역슬래시 등)을 직접 텍스트에 써넣지 않도록
35: 주의해야 한다. 이 글자들을 그 형태 그대로 얻으려면 다음과 같이
36: 타이프한다:  \$  \&  \#  \%  \_  \{  and  \}.
37: 그밖의 기호문자를 어떻게 사용하는지는 이 안내서가 설명해줄 것이다.
38:
39: \end{document}                       % 입력 파일은 이 행으로 끝난다.
```

그림 3: 샘플 LaTeX 파일

1 단순한 문장

단어들은 하나 또는 그 이상의 스페이스로 나누어진다. 단락은 하나 또는 그 이상의 빈 줄로 나누어진다. 출력결과는 입력 파일에 빈 스페이스나 빈 줄을 얼마든지 써넣어도 달라지지 않는다.

겹따옴표는 이렇게 타이프한다: "따옴표친 문구". 홑따옴표는 이렇게 타이프한다: '홑따옴표친 문구'.

긴 대시는 세 개의 대시 문자를 써넣는다—이런 식으로.

이탤릭체 텍스트는 이렇게 한다: *이탤릭체 텍스트*. 굵은 글꼴 텍스트는 이렇게 한다: **굵은 글꼴 텍스트**.

1.1 간단한 경고

문장의 끝이 아닌 위치에 찍은 마침표 다음에 나오는 공백이 너무 큰 경우—예를 들면 etc. 와 같은 일반적인 약자를 쓰는 경우 따위—에는, 마침표 뒤에 역슬래시 하나와 빈 스페이스 하나를 두라. 이 문장을 보면 알 수 있을 것이다.

열 개의 특별한 문자들(예컨대 달러 기호나 역슬래시 등)을 직접 텍스트에 써넣지 않도록 주의해야 한다. 이 글자들을 그 형태 그대로 얻으려면 다음과 같이 타이프한다: \$ \& \# \% _ \{ and \}. 그밖의 기호문자를 어떻게 사용하는지는 이 안내서가 설명해줄 것이다.

그림 4: 샘플 파일의 처리 결과