

The Not So short
Introduction to \LaTeX 2 ϵ

\LaTeX 2 ϵ 입문

142분 동안 익히는 \LaTeX 2 ϵ

by Tobias Oetiker

Hubert Partl, Irene Hyna and Elisabeth Schlegl

Version 4.17, September 27, 2005

김강수, 이기황, MIKA, 김지운, 샘처럼 옮김

4.17-kr, 2005년 11월 5일

Copyright ©1995-2002 Tobias Oetiker and all the Contributors to LShort. All rights reserved.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

이 문서는 프리(free)입니다. 자유 소프트웨어 재단(FSF)에 의해 제출된 GNU GPL(일반 공개 라이선스) 제2판 또는 그 이후 버전이 정하는 바에 따라 자유롭게 재배포하고 수정할 수 있습니다.

이 문서는 유용하게 쓰이기를 바라는 마음으로 배포합니다. 그러나 아무런 보증도 하지 않습니다. 심지어 상업성이나 특정 목적에 적합하다는 보증도 하지 않습니다. 자세한 사항은 GNU GPL을 참조하십시오.

이 문서와 함께 GNU GPL 사본을 받으셨을 것입니다. 그렇지 않다면 Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA로 연락하십시오.

한국어판 저작권

이 책자의 한국어판 저작권은 GNU 자유 문서 라이선스(FDL)를 따릅니다. <http://www.gnu.org/copyleft/fdl.html>

감사의 말

이 안내문서에 사용된 내용 가운데 많은 부분이 오스트리아에서 나온 L^AT_EX 2.09 안내서에서 가져온 것이다. 저자들은 다음과 같다.

Hubert Partl <partl@mail.boku.ac.at>
Zentraler Informatikdienst der Universität für Bodenkultur Wien

Irene Hyna <Irene.Hyna@bmwf.ac.at>
Bundesministerium für Wissenschaft und Forschung Wien

Elisabeth Schlegl <noemail>
in Graz

이 독일어 문서에 관심이 있다면 Jörg Knappen이 L^AT_EX 2_ε 용으로 개선한 판본을 다음 위치에서 찾아볼 수 있다. CTAN:/tex-archive/info/lshort/german

이 책을 쓰면서, `comp.text.tex`에 글을 올리는 사람들에게 자문을 구하였는데 많은 분들이 답변을 보내 주었다. 아래에 기록한 분들은 교정, 제안, 자료를 보내주셔서 이 글이 더 좋은 글이 되도록 도와주신 분들이다. 이 글이 현재 모양을 갖추는 데 이들의 도움이 컸다. 이 분들 모두에게 진심으로 감사드리고자 한다. 당연히, 이 책에서 잘못된 부분이 있다면 그것은 순전히 저자의 책임이다. 제대로 기록된 것이 조금이나마 있다면 그건 틀림없이 아래 있는 분들이 내게 몇 자 적어보내 주신 것 중에 있었을 것이다.

Rosemary Bailey, Marc Bevand, Friedemann Brauer, Jan Busa, Markus Brühwiler, Pietro Braione, David Carlisle, José Carlos Santos, Neil Carter, Mike Chapman, Pierre Chardaire, Christopher Chin, Carl Cerecke, Chris McCormack, Wim van Dam, Jan Dittberner, Michael John Downes, Matthias Dreier, David Dureisseix, Elliot, Hans Ehrbar, Daniel Flipo, David Frey, Hans Fugal, Robin Fairbairns, Jörg Fischer, Erik Frisk, Mic Milic Frederickx, Frank, Kasper B. Graversen, Arlo Griffiths, Alexandre Guimond, Andy Goth, Cyril Goutte, Greg Gamble, Frank Fischli, Morten Høgholm, Neil Hammond, Rasmus Borup Hansen, Joseph Hilferty, Björn Hvittfeldt, Martien Hulsen, Werner Icking, Jakob, Eric Jacoboni, Alan Jeffrey, Byron Jones, David Jones, Johannes-Maria Kaltenbach, Michael Koundouros, Andrzej Kawalec, Sander de Kievit, Alain Kessi, Christian Kern, Jörg Knappen, Kjetil Kjernsmo, Maik Lehardt, Rémi Letot, Flori Lambrechts, Axel Liljencrantz, Johan Lundberg, Alexander Mai, Hendrik Maryns, Martin Maechler, Aleksandar S Milosevic, Henrik Mitsch, Claus Malten, Kevin Van Maren, Richard Nagy, Philipp Nagele, Lenimar Nunes de Andrade, Manuel Oetiker, Urs Oswald, Martin Pfister, Demerson Andre Polli, Maksym Polyakov, Hubert Partl, John Reffing, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Risto Saarelma, Hanspeter Schmid, Craig Schlenter, Gilles Schintgen, Baron Schwartz, Christopher Sawtell, Miles Spielberg, Geoffrey Swindale, Laszlo Szathmary, Boris Tobostras, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zacccone, and Mikhail Zotov.

서문

L^AT_EX [1]은 과학 및 수학 문서를 작성하는 데 적당한 조판 시스템으로서 대단히 뛰어난 조판 품질을 얻을 수 있게 한다. 또한 단순한 편지에서 완전한 단행본에 이르기까지 여러 종류의 문서를 만드는 데도 적합하다. L^AT_EX은 T_EX [2]을 조판 엔진으로 사용한다.

이 길지 않은 입문서는 L^AT_EX 2_ε에 대해서 설명한다. 그리고 L^AT_EX을 응용해서 쓰는 방법을 충분히 소개할 생각이다. L^AT_EX 시스템에 대한 완전한 설명을 보려면 [1, 3]을 참조하면 된다.

이 안내서는 여섯 장으로 이루어져 있다.

제 1 장 L^AT_EX 2_ε 문서의 기본 구조를 설명한다. L^AT_EX의 역사에 대해서도 조금 얘기한다. 이 장을 읽고나면 L^AT_EX이 어떻게 동작하는가에 대해 대강 이해할 수 있게 될 것이다.

제 2 장 문서 조판의 세부사항을 다룬다. 필수적인 L^AT_EX 명령과 환경 거의 대부분을 설명한다. 이 장을 읽은 후에 첫번째 문서를 작성할 수 있게 될 것이다.

제 3 장 L^AT_EX으로 수학식을 식자하는 방법을 설명한다. L^AT_EX의 가장 강력한 기능 중 하나인 수식 표현을 많은 예제를 통하여 보여준다. 이 장의 끝에는 L^AT_EX으로 표현할 수 있는 수학 기호 전부를 표로 정리해두었다.

제 4 장 색인과 문헌목록, 그리고 EPS 그림 삽입을 설명한다. 또 pdfL^AT_EX과 몇 가지 간편한 확장 패키지를 이용하여 PDF 문서를 만들고 보여주는 방법에 대해서 소개한다.

제 5 장 L^AT_EX으로 그림을 그리는 방법에 대해 다룬다. 그래픽 프로그램을 이용하여 그림을 그려서 파일로 저장하여 불러들이는 것이 아니라 L^AT_EX 언어로 그림을 표현하는 방법에 대해서 설명한다.

제 6 장 L^AT_EX이 만들어내는 표준 문서 레이아웃을 변경하는, 약간 위험할 수도 있는 내용을 포함하고 있다. L^AT_EX의 아름다운 출력물을 어떻게 하면 엉망이 되거나 (능력에 따라서) 근사하게 바꿀 수 있는가를 알려준다.

각 장을 순서대로 읽는 것이 중요하다. 이 책은 분량이 그다지 많지 않다. 특히 예제를 주의깊게 보아야 한다. 이 책 전체에 걸쳐 나타나는 예제들에 중요한 정보가 담겨 있기 때문이다.

L^AT_EX은 PC나 매킨토시에서 대규모 UNIX나 VMS에 이르기까지 대부분의 컴퓨터에서 실행된다. 대학의 컴퓨터실에는 대개 L^AT_EX이 이미 설치되어 있을 것이다. 현재 자신이 이용하는 시스템에 L^AT_EX이 설치되어 있는지, 어떻게 사용하여야 하는지에 대해서 알고 싶으면 *Local Guide* [5]를 참고한다. 아무리 해도 잘 되지 않으면 이 책자를 읽으라고 권한 사람에게 물어보라. 이 책이 다루는 범위는 L^AT_EX으로 문서를 작성하는 방법에 대해서 알려주고자 하는 것이지, L^AT_EX 시스템을 설치하고 설정하는 데 대한 것이 아니다.

L^AT_EX 관련 자료가 필요하다면 Comprehensive T_EX Archive Network(CTAN) 사이트를 찾아보라. 홈페이지는 <http://www.ctan.org>이다. ftp 아카이브 <ftp://www.ctan.org>를 통하여 모든 패키지를 내려받을 수 있다. 세계적으로 여러 사이트도 여럿 있다. 예를 들면 <ftp://ftp.ktug.or.kr> (한국), <ftp://ctan.tug.org> (미국), <ftp://ftp.dante.de> (독일), <ftp://ftp.tex.ac.uk> (영국) 등이 있다. 그밖에도 자신이 있는 곳에서 가장 가까운 미러 사이트를 찾아볼 수 있을 것이다.

이 책에 CTAN에 대한 참조가 여러 번 나온다. 특히 다운로드하고자 하는 소프트웨어나 문서를 지시할 때 그러하다. 완전한 URL을 기록하지 않고 단지 CTAN:이라고 지시하고 CTAN 트리 상의 위치만을 표시하였다.

자신의 컴퓨터에서 L^AT_EX을 실행하고자 한다면 <CTAN:/tex-archive/systems>를 참고하면 된다.

이 문서에서 추가하거나 삭제하거나 변경해야 할 부분에 대한 의견이 있다면 저자에게 알려주기 바란다. 특히 이 안내서의 내용이 이해하기 쉬운지 더 좋은 설명 방법은 없겠는지에 대한 L^AT_EX 초보자로부터의 제안을 환영한다.

Tobias Oetiker <oetiker@ee.ethz.ch>

Department of Information Technology and
Electrical Engineering, Swiss Federal Institute of Technology

이 문서의 최신판은 <CTAN:/tex-archive/info/lshort>에서 구할 수 있다.

목차

감사의 말	iii
서 문	v
제 1 장 알아두어야 할 기본 사항	1
제 1 절 어떻게 불러야 하는가	1
1.1 T _E X	1
1.2 L ^A T _E X	2
제 2 절 기본 사항	2
2.1 저자, 디자이너, 조판사	2
2.2 레이아웃 디자인	3
2.3 장점과 단점	3
제 3 절 L ^A T _E X 입력 파일	5
3.1 공백	5
3.2 특수문자	5
3.3 L ^A T _E X 명령	6
3.4 주석	7
제 4 절 입력 파일의 구조	7
제 5 절 명령행 작업	8
제 6 절 레이아웃	10
6.1 문서 클래스	10
6.2 패키지	11
6.3 쪽 양식	12
제 7 절 작업 과정에 생성되는 파일들	12
제 8 절 큰 규모의 글쓰기	15
제 2 장 텍스트의 조판	17
제 1 절 텍스트와 언어의 구조	17
제 2 절 줄바꿈과 쪽나눔	19

2.1	단락 정렬	19
2.2	분절	20
제 3 절	미리 정의된 문자열	21
제 4 절	특수 문자와 기호	21
4.1	따옴표	21
4.2	대시와 하이픈	22
4.3	틸데 (~)	22
4.4	도 기호 (o)	23
4.5	유로 통화 기호 (€)	23
4.6	줄임표 (...)	24
4.7	합자	24
4.8	강세 부호와 특수 문자	25
제 5 절	다국어 지원	26
5.1	한국어 지원	28
5.2	포르투갈어 지원	31
5.3	프랑스어 지원	31
5.4	독일어 지원	32
5.5	키릴 문자 지원	33
제 6 절	띄어쓰기 간격	34
제 7 절	제목과 장, 절	35
제 8 절	상호 참조	37
제 9 절	각주	38
제 10 절	강조	38
제 11 절	환경	39
11.1	Itemize, Enumerate, Description 환경	39
11.2	Flushleft, Flushright, Center 환경	40
11.3	Quote, Quotation, Verse 환경	40
11.4	Abstract 환경	41
11.5	Verbatim 환경	41
11.6	Tabular 환경	42
제 12 절	떠다니는 개체	44
제 13 절	폴리는 명령 보호하기	47
제 3 장	수식 조판하기	49
제 1 절	개괄	49
제 2 절	수학 모드에서 묶기	51
제 3 절	수식 구성하기	52
제 4 절	수식의 간격	56
제 5 절	세로 줄맞춤이 필요한 요소들	56

제 6절	허깨비들	58
제 7절	수학 글꼴 크기	59
제 8절	정리, 법칙, ...	60
제 9절	수학 볼드 기호	61
제 10절	수학 기호 일람	63
제 4장	특별한 기능	71
제 1절	EPS 그림 포함하기	71
제 2절	참고문헌	73
제 3절	찾아보기	74
제 4절	멋진 쪽머리글	75
제 5절	Verbatim 패키지	76
제 6절	L ^A T _E X패키지의 다운로드와 설치	77
제 7절	pdfL ^A T _E X사용하기	78
7.1	PDF 문서의 웹 출판	79
7.2	글꼴	80
7.3	그림 포함하기	82
7.4	하이퍼텍스트 링크	82
7.5	링크 관련 문제	85
7.6	책갈피 관련 문제	85
제 8절	DVIPDFM _x 로 PDF 만들기	87
8.1	한글 pdf 책갈피	88
8.2	그림 삽입	89
8.3	한글 글꼴 기타	89
제 9절	beamer class를 이용한 발표문서 작성	90
제 5장	수학 관련 그림 그리기	93
제 1절	개관	93
제 2절	picture 환경	94
2.1	기본 명령	94
2.2	선분	95
2.3	화살표	96
2.4	원	97
2.5	텍스트와 수식	98
2.6	\multiput과 \linethickness 명령	98
2.7	알모양 곡선. \thinlines와 \thicklines 명령	99
2.8	미리 정의한 그림 박스를 재사용하기	100
2.9	2차 베지어 곡선	101
2.10	현수선	102

2.11 특수상대성이론에서 속도	103
제 3 절 \LaTeX pic	103
제 6 장 \LaTeX을 자기에 맞게 바꾸기	107
제 1 절 새로운 명령, 환경, 패키지	107
1.1 새로운 명령	108
1.2 새로운 환경	109
1.3 여분의 공백	109
1.4 \LaTeX 명령행 조작	110
1.5 사용자 패키지	111
제 2 절 글꼴과 크기	111
2.1 글꼴 바꾸기 명령	111
2.2 경고! 경고!	114
2.3 조연	115
제 3 절 간격	115
3.1 행간	115
3.2 문단 모양	116
3.3 수평 간격	117
3.4 수직 간격	117
제 4 절 페이지 레이아웃	118
제 5 절 길이 문제, 몇 가지 더	120
제 6 절 박스	121
제 7 절 선 그리기	124
참고 문헌	125
Index	127

표 목차

1.1 문서 클래스	10
1.2 문서 클래스의 옵션	11
1.3 L ^A T _E X과 함께 배포되는 패키지 몇 가지	13
1.4 L ^A T _E X의 미리 정의된 쪽 양식	13
2.1 여러 가지 유로 기호들	24
2.2 강세 부호와 특수 문자	25
2.3 포르투갈어 문서를 위한 전처리부	31
2.4 프랑스어를 위한 특수 명령어	32
2.5 독일어 특수 문자들	33
2.6 불가리아, 러시아, 우크라이나어	34
2.7 떠다니는 개체 배치 허용 범위	45
3.1 수학 모드 악센트	63
3.2 그리스 소문자	63
3.3 그리스 대문자	63
3.4 이항 관계 연산자	64
3.5 이항 연산자	64
3.6 큰 연산자	65
3.7 화살표	65
3.8 짝맞춤 기호	65
3.9 큰 짝맞춤 기호	65
3.10 기타 기호	66
3.11 수학 기호가 아닌 것	66
3.12 AMS 짝맞춤 기호	66
3.13 AMS 그리스 및 헤브루 문자	66
3.14 AMS 이항 관계 연산자	67
3.15 AMS 화살표	67
3.16 AMS 부정 이항 관계 연산자 및 화살표	68

3.17 AMS 이항 연산자	68
3.18 AMS 기타 기호	69
3.19 수학 알파벳	69
4.1 graphicx 패키지의 key 명칭	72
4.2 Index Key 사용 예제	75
6.1 글꼴	112
6.2 글꼴 크기	112
6.3 클래스별 글꼴 절대 크기	113
6.4 수학 글꼴	113
6.5 T _E X의 길이단위	118
8.1 각 장별 번역자와 교열자	146

그림 목차

1.1	최소한의 L ^A T _E X 파일	8
1.2	실제 학술지 논문의 보기	9
4.1	fancyhdr 설정 예제	76
4.2	beamer 클래스 샘플 코드	91
6.1	Example Package.	111
6.2	페이지 레이아웃 파라미터	119

제 1 장

알아두어야 할 기본 사항

이 장의 첫 부분은 \LaTeX 2_ε의 개념과 역사에 대해 짧막하게 소개한다. 나머지 부분은 \LaTeX 문서의 기본 구조에 초점을 맞춘다. 이 장을 읽고 나면 \LaTeX 의 동작 방식에 대하여 대략 알게 될 것이고 이 책의 나머지 부분을 이해하는 데 필요한 지식을 갖추게 될 것이다.

제 1 절 어떻게 불러야 하는가

1.1 \TeX

\TeX 은 도널드 크누쓰(Donald E. Knuth)가 만든 컴퓨터 프로그램이다 [2]. 텍스트와 수학적식을 조판하기 위해 만들어졌다. 크누쓰는 \TeX 조판 엔진을 1977년부터 작성하기 시작했는데 그 당시 출판업에서 도입되기 시작하던 디지털 인쇄 장치의 잠재적 가능성을 시험해보고자 한 것이었다. 특히 자신의 책과 논문에게까지 영향을 미치고 있던 타이포그래피 품질 저하 경향을 막아보려 하였다. 오늘날 우리가 사용하는 \TeX 은 1982년에 발표되었고, 1989년에 8비트 문자와 다국어룰 사용할 수 있도록 약간 개선되었다. \TeX 은 매우 안정적인 프로그램으로서 어떤 컴퓨터에서든 실행이 되며 버그 없는 프로그램으로 잘 알려져 있다. \TeX 의 버전 번호는 π 의 십진 전개값으로 붙이는데 현재 버전은 3.141592이다.

\TeX 은 “tech”로 발음한다. 여기서 “ch”는 독일어 “Ach”에서의 “ch”,¹ 또는 스코틀랜드어 “Loch”의 “ch”와 같다. 이 “ch”는 원래 그리스어 알파벳 “chi”에

¹독일어에서 “ch”의 발음은 실제 두 가지라서 어떤 사람들은 “Pech”의 “ch”와 같은 가벼운 소리가 더 적절하다고 생각하는 것 같다. 이 문제에 대해 크누쓰는 독일어 위키백과(Wikipedia)에 다음과 같이 썼다. “나는 사람들이 \TeX 을 어떤 식으로 발음하든 상관하지 않는다 ... 독일에서는 모음 a 뒤에 오는 *ch*보다 e 뒤에 나오는 가벼운 *ch*로 읽는 사람이 많으리라 본다. 러시아에서 ‘*tex*’은 많이 쓰이는 단어이고 발음이 ‘*tyekh*’에 가깝다. 그러나 내 생각에 가장 적절한 것은 그리스어의 발음이라 생각되는데 이것은 *ach*, *loch*와 같은 거친 *ch* 소리이다.”

해당하는 X에서 온 것이다. T_EX은 그리스어 *texnologia* (technology)의 첫 음절이다. 아스키 텍스트로 쓸 때는 T_EX을 TeX으로 쓴다.

1.2 L^AT_EX

L^AT_EX은 T_EX 매크로 패키지이다. L^AT_EX을 사용하면 미리 정의된 전문적인 레이아웃을 이용하여 저자 저작을 고품위 타이포그래피로 조판하고 인쇄할 수 있다. L^AT_EX은 레슬리 램포트(Leslie Lamport)에 의하여 처음 작성되었다 [1]. L^AT_EX은 T_EX을 조판 엔진으로 사용한다. 현재 L^AT_EX은 프랑크 미텔바흐(Frank Mittelbach)가 유지하고 있다.

1994년에 프랑크 미텔바흐가 주도하는 L^AT_EX3 팀이 L^AT_EX 패키지를 개정하였다. 여기서 그 동안 누적되어 온 개선요구가 받아들여지고 그보다 몇 해 전 L^AT_EX 2.09가 발표된 이래 산발적으로 쌓여 온 패치 버전을 전부 통합하여 재정비하였다. 새로운 버전을 이전 것과 구별하기 위하여 L^AT_EX 2_ε라고 부른다. 이 책자가 다루는 것이 L^AT_EX 2_ε이다. L^AT_EX 2.09가 아직도 쓰이는 경우가 있으나 이미 오래된 낡은 버전이다.

L^AT_EX은 “Lay-tech” 또는 “Lah-tech”으로 읽는다. 텍스트 환경에서 LaTeX으로 쓴다. L^AT_EX 2_ε는 “Lay-tech two e”라고 읽고 텍스트에서는 LaTeX2e로 쓴다.

제 2절 기본 사항

2.1 저자, 디자이너, 조판사

책을 출판할 때 저자는 타자기로 원고를 작성하여 출판사에 넘겨준다. 그 회사의 복디자이너 한 명이 이 문서의 레이아웃(외양: 문단폭, 글꼴, 제목줄 앞 뒤의 여백 ...)을 결정한다. 디자이너는 자신의 지시사항을 원고에 써 넣은 다음 조판사에게 넘긴다. 조판사는 이 지시사항에 따라 책을 조판한다.

디자이너는 저자가 원고를 쓰면서 염두에 둔 것이 무엇인가를 파악한 다음, 그 원고의 내용과 자신의 전문지식에 입각하여 장의 제목줄, 인용문, 예문, 수식 등등의 레이아웃을 결정하는 것이다.

L^AT_EX을 사용하는 경우, L^AT_EX이 디자이너 역할을 한다. 그리고 T_EX을 조판기로 이용한다. 그러나 L^AT_EX은 컴퓨터 프로그램에 지나지 않으므로, (사람인 디자이너에게 맡기는 경우보다) 더 많은 지시사항을 알려주어야 할 것이다. 저자는 자기 글의 논리적 구조를 나타내는 정보를 텍스트와 함께 추가로 제공하여야 하는데, 이 정보는 “L^AT_EX 명령”이라는 형태로 텍스트에 써넣는다.

이러한 방식은 이른바 WYSIWYG²이라 불리는, MS Word나 아래아한글³

²What you see is what you get.

³원문에는 Corel Word-Perfect —[역주]

등, 요즘의 거의 모든 워드 프로세서들이 취하는 방법과는 상당히 다른 것이다. WYSIWYG 방식은 저자가 컴퓨터에 텍스트를 타이핑하면서 대화식으로 문서의 레이아웃을 조정해간다. 즉 자신의 최종 작업 결과가 출력되었을 때의 모양을 화면으로 보면서 작업할 수 있는 것이다.

L^AT_EX을 사용하면 텍스트를 입력하는 동안 최종 출력물의 모양을 보지는 못한다. 그러나 그 입력 파일을 L^AT_EX으로 처리(컴파일)한 후에는 화면으로 최종 출력결과를 미리볼 수 있다. 문서를 실제 프린트하기 전에 화면으로 결과를 보면서 수정하는 것도 가능하다.

2.2 레이아웃 디자인

타이포그래피 디자인은 전문적인 일이다. 이 분야에 익숙지 않은 저자들이 자주 저지르는 심각한 실수는, 책을 디자인하는 것이 대부분 미관의 문제라고 생각하는 것이다. “디자인이 잘 된 거? 문서가 보기 좋으면 되는 거 아냐?” 그러나, 문서라는 것은 읽기 위해 있는 것이지 화랑에 걸어두려고 있는 것이 아니다. 가독성과 이해가능성은 그 외관의 아름다움보다 훨씬 더 중요하다. 예를 들어보자.

- 독자에게 장·절의 구조가 명확하게 전달되도록 장·절 표제의 글꼴 크기와 번호 붙이기를 정해야 한다.
- 행 길이는 독자의 시야를 넘어서지 않을 정도로 짧아야 하지만, 쪽을 보기 좋게 채울 정도로 길어야 한다.

WYSIWYG 시스템을 사용하는 저자들은 외관상 보기에는 좋을지 몰라도 도대체 구조적이지 못하고 일관성도 없는 문서를 만들어내는 경우가 많다. L^AT_EX은 저자 자신의 문서에 논리적인 구조를 선언하게 함으로써 이러한 구조 오류를 피하게 해준다. 그런 다음 이 구조에 가장 적당한 레이아웃을 L^AT_EX이 선택하는 것이다.

2.3 장점과 단점

WYSIWYG에 익숙한 사람들이 L^AT_EX을 사용하는 사람을 만나거나 하면 “일반 워드 프로세서에 비해 L^AT_EX이 가진 장점이 무엇이나”거나, 반대로 워드 프로세서의 좋은 점이 무엇이나에 대해 논쟁이 벌어지는 일이 종종 있다. 이런 논쟁이 벌어지면 자리를 피하는 것이 상책이다. 왜냐하면 이런 식의 논의는 건잡을 수 없게 비화하기 일쑤이기 때문인데, 그래도 가끔은 피할 수 없을 때가 있는 법이라서….

그래서 여기 탄약 몇 발을 마련해둔다. L^AT_EX이 일반 워드 프로세서에 비해서 더 좋은 점은 다음과 같다.

- 전문가 수준으로 디자인된 레이아웃을 사용할 수 있다. 그래서 문서가 “인쇄된” 것과 거의 동일하다.
- 수학적 조판이 매우 쉽다.
- 사용자는 문서의 논리적 구조를 지시하는 몇 가지 기억하기 쉬운 명령어들만 익히면 된다. 이 명령들은 대부분 눈으로 보이는 문서의 모양을 주물러야 하는 것이 아니다.⁴
- 각주, 상호참조, 목차, 참고문헌 등 매우 복잡한 구조들도 아주 쉽게 만들어진다.
- L^AT_EX 기본 패키지 이외에도 여러 가지 조판상의 필요를 충족시키는 자유롭게 이용할 수 있는 추가 패키지들이 많다. POSTSCRIPT 그림을 포함하게 하거나 표준을 정확하게 지키는 참고문헌을 작성하게 하거나 하는 패키지들을 예로 들 수 있겠다. 이 추가 패키지들은 대부분 *The L^AT_EX Companion* [3]에 설명되어 있다.
- L^AT_EX은 구조화가 잘 된 문서를 작성하도록 저자를 유도한다. L^AT_EX의 동작 방식이 바로 그것이다—구조를 지정하는 것.
- L^AT_EX 2_ε의 조판 엔진인 T_EX은 매우 이식성이 높으며 자유 소프트웨어이다. 그러므로 T_EX 시스템은 거의 모든 하드웨어 플랫폼에서 실행 가능하다.

L^AT_EX에는 단점도 있다. 사실 나로서는 문제거리라고 할 만한 것을 찾기가 어렵지만, 그런 단점을 수백 가지씩 지적해 내는 사람도 틀림없이 있는 것이다. ;-)

- L^AT_EX은 아무 생각 없는 머리가 빈 사람들하고는 잘 맞지 않는다...
- 미리 만들어진 레이아웃 인자의 기정값을 조절해가면서 쓸 수 있기는 하지만, 완전히 새로운 레이아웃을 설계하는 것은 어렵고 시간이 많이 드는 일이다.⁵
- 구조화되지 않고 비체계적인 문서를 작성하기가 너무나 어렵다.
- 햄스터에게는 아무리 노력해도 논리적 마크업의 개념을 이해시키기 쉽지 않을 것이다.

⁴즉, 논리적 구조를 선언하는 것이지 시각적 모양을 조절하는 것이 아니라는 뜻. —[역주]

⁵소문에 의하면 앞으로 나올 L^AT_EX3 시스템에서는 이것이 주안점이 될 거라 한다.

제 3 절 L^AT_EX 입력 파일

L^AT_EX에 의하여 처리되는 것은 플레인 텍스트 파일(plain ASCII text)이다. 어떤 텍스트 편집기(에디터)로도 만들 수 있다. 여기에는 문서의 내용은 물론이고, L^AT_EX에게 텍스트를 조판하는 방법에 관하여 알려주는 지시사항들도 기록된다.

3.1 공백

“공백문자(whitespace characters)”, 즉 빈 칸(blank), 탭(tab) 등은 L^AT_EX에서 모두 동일하게 “스페이스”로 처리된다. 계속되는 여러 개의 공백문자들은 하나의 “스페이스”로 취급된다. 행의 첫머리에 있는 공백문자들은 무시되고 한번의 줄바꿈(개행:linebreak) 역시 “공백문자”로 간주된다.

두 줄 사이에 빈 줄을 하나 넣으면 문단(paragraph)의 끝을 나타낸다. 빈 줄을 여러 개 넣어도 한 줄을 넣은 것과 효과가 같다. 다음 예를 보자. 왼쪽은 입력 파일에 적힌 텍스트의 내용이고 오른쪽은 그 출력된 형태를 보여준다.

```
It does not matter whether you
enter one or several      spaces
after a word.
```

```
An empty line starts a new
paragraph.
```

It does not matter whether you enter one or
several spaces after a word.

An empty line starts a new paragraph.

3.2 특수문자

다음 기호들은 유보된 문자들로서 L^AT_EX에서 특별한 의미를 갖거나 어떤 글꼴로도 나타낼 수 없는 것들이다. 이것을 텍스트에 직접 써넣는다면 대개 프린트되지 않을 것이며 그럼에도 L^AT_EX에게 강제로 실행하도록 하면 원치 않는 결과를 초래할 것이다.

\$ % ^ & _ { } ~ \

앞으로 보게 되겠지만, 이러한 문자를 문서에서 사용하려면 백슬래시(\)를 더해 주어야 한다.

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

\$ % ^ & _ { } ~

위의 보기에서 종괄호{ }를 붙인 명령이 두어 개 있는데 이것은 원래 그 뒤에 오는 문자에 강제 부호를 붙이는 데 사용되는 것이기 때문에 부호만을 표기하기 위하여 이런 방식으로 입력하는 것이다. 강제 부호에 대해서는 25 쪽에서 자세히 설명한다.

그밖의 많은 기호와 부호, 수식에서 쓰이는 문자, 억양 표시가 붙은 문자들은 특별한 명령을 써서 출력한다. 백슬래시 문자 \는 백슬래시를 그 앞에 하나 더 붙인다고(\\) 얻어지는 것이 아니다. 이 \\ 부호는 줄바꿈(개행)에 사용된다.⁶

3.3 L^AT_EX 명령

L^AT_EX 명령은 대소문자를 구별하며 다음과 같은 두 가지 형태 가운데 하나를 취한다.

- 백슬래시 \ 로 시작하여 문자(letter)만으로 이루어진 이름을 갖는다. 명령 이름은 공백이나 숫자 또는 ‘문자 아닌 것’이 오면 끝난다.
- 백슬래시 다음에 딱 한 개의 기타 문자(non-letter)가 온다.

L^AT_EX은 명령 다음의 공백문자를 무시한다. 명령 다음에 빈 칸을 두고 싶으면 {}와 빈 칸을 써넣거나 명령 이름 뒤에 특별한 칸 띄우기 명령을 써야 한다. {}는 L^AT_EX이 명령 이름 다음에 나오는 빈 칸들을 “먹어버리지”(무시하지) 않도록 만든다.

```
I read that Knuth divides the
people working with \TeX{} into
\TeX{}nicians and \TeX perts.\\
Today is \today.
```

```
I read that Knuth divides the people working
with TEX into TEXnicians and TEXperts.
Today is 2005년 11월 5일.
```

인자(parameter)가 필요한 명령도 있다. 인자는 명령 이름 다음의 중괄호 { } 속에 써넣어야 한다. 어떤 명령에는 옵션 인자(optional parameters)가 필요한 경우도 있는데, 명령 바로 다음에 대괄호 []를 쓰고 그 안에 써넣는 것이 이것이다. 다음은 L^AT_EX 명령을 사용하는 보기이다. 모르는 명령이 있어도 걱정할 것 없다. 나중에 설명이 될 것이다.

```
You can \textsl{lean} on me!
```

```
You can lean on me!
```

```
Please, start a new line
right here!\newline
Thank you!
```

```
Please, start a new line right here!
Thank you!
```

⁶ ‘\’를 얻으려면 `\backslash` 또는 `\textbackslash` 명령을 사용한다.

3.4 주석

L^AT_EX은 입력 파일을 처리해가다가 % 문자를 만나면 그 줄(행)의 나머지 부분과 줄바꿈을 무시한다. 그리고 다음 줄 첫머리에 오는 공백문자도 모두 무시한다.

이것은 입력 파일 작성시 최종 인쇄판에는 나타나지 않는 주석문을 쓰고 싶을 때 사용할 수 있다.

```
This is an % stupid
% Better: instructive <----
example: Supercal%
         ifragilist%
         icexpialidocious
```

This is an example: Supercalifragilisticexpialidocious

% 문자는 공백문자나 줄바꿈 문자가 허용되지 않는 긴 줄을 나누어 입력할 수 있게 해준다.

좀 더 긴 주석문을 쓰려면 `verbatim` 패키지가 제공하는 `comment` 환경을 쓸 수 있다.⁷ 이 말은 `comment` 환경을 사용하려면 문서의 전처리부(preamble)에 `\usepackage{verbatim}` 이라는 명령 한 줄을 추가해야 한다는 뜻이다.

```
This is another
\begin{comment}
rather stupid,
but helpful
\end{comment}
example for embedding
comments in your document.
```

This is another example for embedding comments in your document.

이 환경은 수식과 같은 복잡한 환경 안에서는 잘 작동하지 않을 수 있다는 점에 주의하라.

제 4 절 입력 파일의 구조

L^AT_EX이 입력 파일을 처리할 때는 그 파일이 일정한 구조를 따르고 있다고 가정한다. 그러므로 모든 입력 파일은 다음 명령으로 시작해야 한다.

```
\documentclass{...}
```

이 명령은 지금 작성하고자 하는 문서가 어떤 종류의 것인지 설정하는 것이다. 이 다음에 전체 문서의 모양(스타일)에 영향을 주는 명령들을 포함하거나 L^AT_EX 시스템에 새로운 기능을 추가하는 패키지들을 포함할 수도 있다. 패키지를 포함할 때는 다음과 같은 형태의 명령을 쓴다.

⁷comment 패키지도 `comment` 환경을 제공한다. —[역주]

```
\usepackage{...}
```

이 설정 과정이 모두 끝난 뒤에,⁸ 문서의 주요부가 시작된다. 다음이 문서 주요부의 시작을 나타내는 명령이다.

```
\begin{document}
```

이제 문장을 몇 가지 L^AT_EX 명령과 섞어가면서 입력하면 된다. 문서의 끝에는

```
\end{document}
```

명령을 추가한다. 이것은 L^AT_EX에게 처리해야 하는 부분의 끝임을 알려준다. 이 명령 이후에 나오는 것은 모두 무시된다.

그림 1.1에 보인 것은 최소한의 L^AT_EX 파일 내용이다. 이보다 좀 복잡한 입력 파일이 그림 1.2에 있다.

```
\documentclass{article}
\begin{document}
Small is beautiful.
\end{document}
```

그림 1.1: 최소한의 L^AT_EX 파일

제 5 절 명령행 작업

독자는 틀림없이 이 작은 L^AT_EX 입력 파일을 가지고 무얼 어찌해야 좋을지 몰라서 당황하고 있을 것이다. 여기 도움말을 조금 적어둔다. L^AT_EX 자체는 GUI(그림 사용자 인터페이스)나 예쁜장한 누름 버튼이 없는 프로그램이다. L^AT_EX은 그저 입력 파일을 정해진 규칙에 따라 차례로 처리하여 출력 파일을 만들어낼 뿐이고 사용자에게 어떤 시각적 즐거움을 주는 동작을 하지 않는다. L^AT_EX이 설치된 시스템 중에는 그림(graphic) 기반 환경을 제공하여 L^AT_EX 버튼을 누르면 입력 파일을 컴파일할 수 있는 경우도 있다. 그렇지만 명령행에 명령을 써넣는 전통적인 시스템도 있으므로 여기서는 이러한 텍스트 기반 시스템에서 입력 파일을 컴파일하도록 L^AT_EX을 부르는 방법을 알아보자. 당연히 L^AT_EX 설치에 이미 잘 되어 있는 시스템임을 가정한다.⁹

1. L^AT_EX 입력 파일을 만들거나 편집한다. 이 파일은 플레인 텍스트 파일이어야 한다. 유닉스 시스템의 모든 에디터는 플레인 텍스트 파일을 만

⁸\documentclass와 \begin{document} 사이에 오는 부분을 전처리부(*preamble*)라 한다.

⁹대부분의 Unix 시스템의 경우가 이에 해당한다. 말하자면... 진짜배기는 Unix를 사용하는 법, 그러니까... ;-))

```

\documentclass[a4paper,11pt]{article}
% define the title
\author{H.~Partl}
\title{Minimalism}
\begin{document}
% generates the title
\maketitle
% insert the table of contents
\tableofcontents
\section{Some Interesting Words}
Well, and here begins my lovely article.
\section{Good Bye World}
\ldots{} and here it ends.
\end{document}

```

그림 1.2: 실제 학술지 논문의 보기

든다. 윈도우에서라면 파일을 ASCII 또는 *plain text* 포맷으로 저장해야 할 것이다. 파일 이름은 확장명을 *.tex*으로 붙인다.

2. 입력 파일에 대해서 \LaTeX 을 실행한다. 성공하면 *.dvi* 파일을 얻게 될 것이다. 목차나 내부 상호참조를 완성하려면 \LaTeX 을 두세번 실행해야 할 것이다. 입력 파일에 오류가 있으면 \LaTeX 이 이를 알려주고 실행을 멈춘다. Ctrl-D를 눌러서 명령행으로 돌아온다.

```
latex foo.tex
```

3. 이제 DVI 파일을 미리보기할 수 있다. 방법은 여러 가지가 있는데 스크린으로 파일을 보려면 예컨대 다음과 같이 실행한다.

```
xdvi foo.dvi &
```

이것은 X Window가 실행되는 Unix에만 해당한다. 윈도우라면 (MiKTeX에서 제공하는) *yap* (Yet Another Previewer)과 같은 프로그램이 있다.

dvi 파일을 POSTSCRIPT로 바꾸어서 인쇄하거나 Ghostscript 프로그램으로 화면보기할 수 있다.

```
dvips -Pcmz foo.dvi -o foo.ps
```

\LaTeX 시스템에 *dvipdf* 유틸리티가 설치되어 있는 경우라면 *.dvi* 파일

을 직접 pdf로 변환할 수도 있을 것이다.

```
dvipdf foo.dvi
```

제 6 절 레이아웃

6.1 문서 클래스

L^AT_EX에게 입력 파일을 처리하게 할 때 첫번째로 제공하여야 하는 정보는 저자가 만들려는 문서의 유형(type)이다. 이것은 `\documentclass` 명령으로 지정한다.

```
\documentclass[options]{class}
```

여기서 *class*라는 인자는 만들어질 문서의 유형을 지시한다. 표 1.1에 이 책에서 설명하는 문서 클래스들을 요약해 놓았다. L^AT_EX 2_ε 배포본에는 편지나 슬라이드 등의 문서를 위한 클래스가 더 포함되어 있다. *option* 인자는 문서 클래스의 동작을 사용자의 의도에 맞게 조절하는 데 쓰인다. 여러 개의 옵션은 쉼표로 구분한다. 표준적 문서 클래스에 사용되는 일반적 옵션을 표 1.2에 요약하였다.

예를 들어보자. 입력 파일의 첫 줄이 다음과 같이 시작되었다면,

표 1.1: 문서 클래스

article 과학 학술지, 프리젠테이션, 짧은 보고서, 프로그램 문서, 초대장 등에 쓰이는 아티클용 클래스

proc article 클래스에 기초한 프로시딩을 위한 클래스

minimal 최소 문서 양식 클래스. 페이지 크기와 기본 글꼴만을 설정한다. 주로 디버깅을 위하여 사용함.

report 여러 장(chapter)으로 이루어진 긴 보고서, 작은 책, 박사학위 논문 등에 쓰이는 클래스.

book 진짜 책을 만들기 위한 클래스.

slides 슬라이드 제작용 클래스. 큰 산세리프 글꼴을 사용한다. 이것 대신 FoilT_EX의 사용도 고려해볼 수 있다.^a

^amacros/latex/contrib/foiltex

표 1.2: 문서 클래스의 옵션

10pt, 11pt, 12pt	문서 기본 글꼴 크기를 설정한다. 아무 옵션도 주지 않으면 10pt로 간주된다.
a4paper, letterpaper, ...	종이 크기를 지정한다. letterpaper가 기본 종이 크기이다. 이외에도 a5paper, b5paper, executivepaper, legalpaper 등을 선택할 수 있다.
fleqn	디스플레이 수식을 가운데 정렬이 아닌 왼쪽 정렬로 식자한다.
leqno	수식 번호를 수식의 오른쪽이 아닌 왼쪽에 표시되도록 한다.
titlepage, notitlepage	표지 뒤에 새로운 페이지를 시작하도록 할 것인지 지정한다. article 클래스는 새 페이지를 시작하지 않는 것이 기본값이고 report와 book은 새 페이지를 만든다.
onecolumn, twocolumn	문서를 1단 또는 2단으로 조판하도록 지시한다.
twoside, oneside	양면인쇄용 또는 단면인쇄용 출력물을 만들라고 지시하는 것이다. article과 report는 단면, book 클래스는 양면이 기본값이다. 이 옵션은 문서의 모양에 관한 것일 뿐이다. 즉, twoside 옵션을 주었다고 해서 프린터가 실제로 양면으로 출력해주는 것은 아니다.
landscape	레이아웃을 가로가 긴 형식(landscape)으로 변경한다.
openright, openany	새로운 장을 홀수쪽에서만 시작할 것인지 홀/짝 구분 없이 바로 다음 쪽에서 시작할 것인지를 지정하는 것이다. 이 옵션은 장(chapter)이라는 개념이 없는 article 클래스에서는 동작하지 않는다. report 클래스는 홀·짝 구분 없이 다음 쪽에서 시작하는 것이 기본값으로 되어 있고 book 클래스에서는 홀수쪽에서 시작하는 것이 기본값이다.

```
\documentclass[11pt,twoside,a4paper]{article}
```

이것은 L^AT_EX에게 문서를 *article*로 작성하고 기본 글꼴 크기를 11포인트로 하고, A4 용지에 양면 인쇄하기에 적당하도록 레이아웃을 구성하라는 것을 지시하고 있다.

6.2 패키지

글을 써가다 보면 기본 L^AT_EX만으로는 해결할 수 없는 문제를 만날 때가 있을 것이다. 그림을 포함하려 하거나 채색된 글씨를 쓰고 싶을 때, 또는 문서에 소스 코드를 삽입해야 할 경우, L^AT_EX의 기능을 향상시켜야 할 필요를 느끼게 된다. 이러한 기능 향상은 패키지를 통해서 이루어진다. 다음 명령을 써서

패키지를 사용 가능하게 한다.

```
\usepackage[options]{package}
```

여기서 *package*란 패키지의 이름을 가리키고, *options*란 패키지가 수행해야 할 특정 기능을 나타내는 지시어들을 말한다. L^AT_EX 2_ε 기본 배포판에 함께 따라오는 패키지들도 있다. (표 1.3을 참조.) 그밖에 독자적으로 제공되는 패키지도 있다. 자신의 컴퓨터에 어떤 패키지들이 설치되어 있는지 알고 싶으면 거기에 있는 *Local Guide* [5]를 찾아보면 된다. L^AT_EX 패키지에 대해서는 *The L^AT_EX Companion* [3]을 보면 많은 것을 알 수 있을 것이다. 이 책은 여러 패키지에 대한 설명은 물론이고 L^AT_EX 2_ε를 확장하기 위해 자신이 직접 패키지를 작성하는 방법에 대해서도 설명하고 있다.

요즘 T_EX 배포판은 수많은 패키지를 함께 설치하여 준다. Unix 시스템에 서라면 `texdoc` 명령을 사용하여 패키지 문서를 쉽게 열어볼 수 있다.

6.3 쪽 양식

머리글(header)과 바닥글/footer)의 조합을 쪽 양식(page style)이라 한다. L^AT_EX은 세 가지 미리 정의된 쪽 양식을 제공한다.

```
\pagestyle{style}
```

위의 명령으로 설정할 수 있는 기정의 쪽 양식은 표 1.4에 요약되어 있다. 현재 쪽만 쪽 양식을 바꾸려면 다음 명령을 사용한다.

```
\thispagestyle{style}
```

사용자가 스스로 머리글과 바닥글을 만드는 방법은 *The L^AT_EX Companion* [3]에도 잘 나와 있고 이 책의 75 페이지 제 3장 제 4절에 설명되어 있다.

제 7절 작업 과정에 생성되는 파일들

L^AT_EX으로 작업하게 되면 곧 여러 가지 확장명을 가진 파일에 혼란을 느끼게 된다. 다음 목록은 T_EX으로 작업할 때 만나게 되는 파일 유형을 설명한 것이다. 이 목록이 모든 확장명을 모두 설명하는 것은 아님을 기억하라. 만약 중요한데도 빠진 것이 있다면 저자에게 알려주기 바란다.

.tex T_EX 또는 L^AT_EX 입력 파일. latex으로 컴파일할 수 있다.

.sty L^AT_EX 매크로 패키지. \usepackage 명령으로 L^AT_EX 문서에 불러올 수 있다.

표 1.3: L^AT_EX과 함께 배포되는 패키지 몇 가지

doc	L ^A T _E X 프로그램을 문서화할 수 있게 한다. doc.dtx ^a 와 <i>The L^AT_EX Companion</i> [3]에 설명되어 있다.
exscale	확장 수학 글꼴을 크기별로 조절하는 기능을 제공한다. ltxscale.dtx에 설명되어 있다.
fontenc	L ^A T _E X에서 사용되는 폰트 인코딩을 지정한다. loutenc.dtx에 설명되어 있다.
ifthen	‘if...then do...otherwise do...’ 형식의 조건 명령을 작성할 수 있게 한다. ifthen.dtx와 <i>The L^AT_EX Companion</i> [3]에 설명되어 있다.
latexsym	L ^A T _E X 기호문자(symbol) 글꼴을 식자하려면 latexsym 패키지를 사용해야 한다. latexsym.dtx와 <i>The L^AT_EX Companion</i> [3]에 설명되어 있다.
makeidx	색인을 작성하는 명령을 제공한다. 74 페이지 제 3절과 <i>The L^AT_EX Companion</i> [3]에 설명되어 있다.
syntonly	문서를 처리(컴파일)하지만 조판결과는 만들지 않는다.
inputenc	입력 인코딩을 지정할 수 있도록 한다. ASCII, ISO Latin-1, ISO Latin-2, IBM 코드 페이지 437/850, 애플 매킨토시, 넥스트, ANSI-윈도우, 사용자 정의 등의 인코딩을 선택할 수 있다. inputenc.dtx에 설명되어 있다.

^a이 파일은 시스템에 이미 설치되어 있을 것이다. 쓰기 권한이 있는 아무 디렉토리에서나 latex doc.dtx라고 타이핑하면 dvi 문서를 얻을 수 있다. 이 표에 나오는 다른 파일에 대해서도 마찬가지이다.

표 1.4: L^AT_EX의 미리 정의된 쪽 양식

plain	쪽 번호를 쪽의 아래쪽 바닥글에 중앙정렬하여 찍는다. 쪽 양식의 기본값이다.
headings	현재 장 표제와 쪽 번호를 각 쪽의 머리글에 적는다. 바닥글은 비운다. (이 문서에서 사용된 양식이다.)
empty	머리글과 바닥글을 모두 비운다.

`.dtx` 문서화된 \TeX 파일. \LaTeX 스타일 파일은 이 형식으로 배포된다. `.dtx` 파일을 컴파일하면 `.dtx` 속에 들어 있는 \LaTeX 패키지의 매크로 코드와 문서를 얻을 수 있다.

`.ins` `.dtx` 파일에 들어있는 파일들을 풀어내기 위한 설정 파일. 인터넷에서 \LaTeX 패키지를 내려받았다면 대개 `.dtx`와 `.ins` 파일로 되어 있을 것이다. `.ins` 파일에 \LaTeX 을 실행하면 `.dtx` 파일을 풀어준다.

`.cls` 문서 기본 형태를 정의하는 클래스 파일. `\documentclass` 명령으로 선택한다.

`.fd` \LaTeX 에게 새 글꼴에 대해 알려주는 폰트 정의 파일.

다음 파일은 \LaTeX 으로 입력 파일을 컴파일할 때 생성되는 것들이다.

`.dvi` 장치 독립(device independent) 파일. \LaTeX 의 주 출력 형식이다. DVI 미리보기 프로그램으로 내용을 볼 수 있고 `dvips`와 같은 응용 프로그램으로 인쇄할 수 있다.

`.log` 직전의 컴파일 과정에서 무슨 일이 일어난 것인지를 상세하게 알려주는 로그 파일.

`.toc` 장절 표제를 저장해 두는 파일. 다음 번 컴파일 때 \LaTeX 이 읽어서 목차를 만드는 데 사용한다.

`.lof` `.toc`와 같지만 그림 목차에 해당한다.

`.lot` 표 목차를 위한 파일이다.

`.aux` 컴파일러는 필요한 정보를 이 파일에 적어서 다음 번 실행 때 읽어들이도록 한다. 주로 상호참조를 완성하기 위해 `.aux` 파일이 주로 쓰인다.

`.idx` 문서에 색인을 넣어야 한다면 \LaTeX 은 파일의 모든 색인 지시자에 해당하는 단어들을 모아야 한다. 이것이 `.idx` 파일이다. 이렇게 모인 단어들을 `makeindex`로 처리하여 색인을 만들게 된다. 색인 만들기에 대한 좀더 자세한 사항은 74 페이지의 제 3 절을 보라.

`.ind` `.idx` 파일을 처리하여 다음 번 컴파일 시에 문서에 포함할 색인 파일이다.

`.ilg` `makeindex` 실행시의 로그 파일이다.

제 8 절 큰 규모의 글쓰기

큰 문서를 작업하려면 입력 파일을 몇 개 부분으로 나누어 작업하고 싶을 때가 있다. L^AT_EX은 이를 위해 두 가지 명령을 제공한다.

```
\include{filename}
```

이 명령을 사용하면 문서의 일부에 *filename.tex*이라는 다른 파일의 내용을 삽입할 수 있다. L^AT_EX은 *filename.tex*의 내용을 삽입하기 전에 새로운 쪽을 시작한다는 점에 주의하라.

`\includeonly`라는 또 하나의 명령은 전처리부(preamble)에서만 쓸 수 있다. `\include` 명령으로 파일을 지정했을 때 그 파일이 여기 열거된 것 중 하나일 때만 삽입하도록 L^AT_EX에게 지시하는 것이다.

```
\includeonly{filename,filename,...}
```

이 명령이 문서의 전처리 부분에서 실행된 후에는 `\include` 명령으로 삽입하는 파일들 가운데 여기서 지정된 것만이 실제로 삽입될 것이다. 파일 이름과 쉼표 사이에 공백이 없어야 한다는 데 주의한다.

`\include` 명령은 파일을 삽입할 때 새로운 쪽을 만들고 거기에서부터 조판하기 시작한다. `\includeonly`를 쓰는 경우 이렇게 하는 것이 유리하다. 왜냐하면 `\include`된 파일 중에서 몇 개를 제외하더라도 쪽 나눔 위치가 이동하지 않기 때문이다. 가끔 포함되는 파일 앞에서 쪽을 나누는 것이 바람직하지 못할 때도 있다. 이럴 경우

```
\input{filename}
```

명령을 쓰면 된다. 이 명령은 다만 지정된 파일을 명령이 내려진 위치에서 불러들일 뿐이고 쪽 조절이나 코드 추가를 하지 않는다.

문서의 구문만을 검사하고 싶으면 `syntonly` 패키지를 사용한다. L^AT_EX은 문서를 훑어보면서 명령의 구문과 사용법이 적절한지만을 검사하고 출력물(DVI)을 만들지 않는다. L^AT_EX은 이 모드에서 훨씬 빠르게 동작하므로 시간을 절약할 수 있다. 간단히 다음과 같이 하면 된다.

```
\usepackage{syntonly}
\syntonly
```

DVI 출력을 만들고 싶으면 위의 두번째 줄을 주석 처리(퍼센트 기호를 줄 앞에 붙임)하라.

제 2 장

텍스트의 조판

앞 장을 다 읽은 독자들은 기본적인 $\text{\LaTeX} 2_{\epsilon}$ 문서 작성법을 알게 되었을 것이다. 이 장에서는 실제로 사용 가능한 제대로 된 문서를 작성하기 위해 필요한 문서의 구조적 요소 등에 관하여 알아본다.

제 1 절 텍스트와 언어의 구조

글쓰기의 주목적은 (현대적인 몇몇 특이한 문학 작품들은 빼놓고) 글쓴이의 사상과 정보 혹은 지식을 독자들에게 전달하는 데에 있다. 내용이 잘 구성된 글은 훨씬 이해하기가 좋다. 나아가 문서의 조판 형식이 문서 내용의 논리적·의미적 구조를 잘 반영하고 있다면 독자에게 더 큰 도움이 될 것이다.

\LaTeX 은 텍스트의 논리적·의미적 구조만 규정하면 된다는 점에서 다른 문서 조판 시스템들과 구분이 된다. \LaTeX 은 문서 클래스 파일과 여러 스타일 파일에 주어진 “규칙”에 따라 텍스트의 조판 형식을 만들어 낸다.

\LaTeX 과 문서 조판에 있어서 가장 중요한 텍스트 단위는 단락이다. 단락을 “텍스트 단위”라고 부르는 것은 단락이 한 덩이의 일관된 생각이나 개념을 반영해야 하는 조판 형식이기 때문이다. 다음 절에서 `\`를 이용하여 줄바꿈을 하는 법, 원본 입력 파일에 빈 줄을 두어서 단락을 나누는 법을 배우게 될 것이다. 따라서 새로운 생각이 시작되는 곳에서는 새로운 단락이 시작되어야 하고, 그렇지 않은 경우에는 줄바꿈만을 하여야 한다. 단락 나눔에 관하여 확신이 서지 않으면 자신의 텍스트를 사상과 사고의 전달 장치라는 관점에서 살펴보자. 이전의 생각이 이어지고 있는데도 단락을 나누었다면 다시 합쳐야 한다. 같은 단락 안에서 완전히 새로운 생각이 나온다면 그 곳에서 단락을 나누어야 한다.

대부분의 사람들이 단락 나누기를 별로 중요하지 않게 생각한다. 많은 사람들이 단락 나눔의 의미조차 모르고 있다. 또한 \LaTeX 사용자 가운데에는

단락 나눔을 아무 생각없이 하는 사람들도 있다. 후자의 실수는 텍스트에서 수식이 사용된 경우 자주 발생한다. 다음 예를 살펴보면서 어떤 경우에 수식의 앞뒤에 빈 줄(단락 나눔)이 사용되고, 또 어떤 경우에는 사용되지 않는가를 생각해 보라.(이 예에 사용된 명령들을 다 알지 못해서 이해가 힘들면 이 장과 다음 장을 모두 읽은 다음 이 절을 다시 읽기 바란다.)

```
% Example 1
\ldots when Einstein introduced his formula
\begin{equation}
e = m \cdot c^2 \; ,
\end{equation}
which is at the same time the most widely known
and the least well understood physical formula.
```

```
% Example 2
\ldots from which follows Kirchhoff's current law:
\begin{equation}
\sum_{k=1}^n I_k = 0 \; .
\end{equation}
```

```
Kirchhoff's voltage law can be derived \ldots
```

```
% Example 3
\ldots which has several advantages.
```

```
\begin{equation}
I_D = I_F - I_R
\end{equation}
is the core of a very different transistor model. \ldots
```

단락 다음 단계의 텍스트 단위는 문장이다. 영문 텍스트에서는 문장의 끝을 나타내는 마침표 뒤에 오는 공백이 약어를 표시하는 마침표 뒤에 오는 공백보다 크다. \LaTeX 은 마침표가 어떤 의도로 사용되었는가를 스스로 판단하려 한다. 만일 \LaTeX 의 판단이 잘못되었다면 글쓴이가 직접 원하는 바를 지시해 주어야 한다. 이에 관해서는 이 장의 후반부에서 다룬다.

텍스트의 구조화는 문장의 부분 요소에까지 이른다. 문장 부호 사용법은 대부분의 언어에서 매우 복잡하다. 그러나 많은 경우 (독일어와 영어를 포함해서) 쉼표가 말의 흐름에서 짧은 쉼을 나타낸다는 것을 기억하면 쉼표를 어디에다가 찍을 것인가는 어렵지 않게 판단할 수 있다. 어디에다 쉼표를 찍을지 잘 모르겠으면 그 문장을 크게 읽으면서 쉼표가 있는 곳에서마다 짧게 숨을 쉬어 보라. 숨을 쉬기가 어색한 곳이 있으면 그 곳에 있는 쉼표를 지우

도록 하라. 혹은 다른 곳에서 숨을 쉬거나 잠시 멈추는 좋겠다는 느낌을 들면 그 곳에 쉼표를 찍으면 된다.

문장이 모여서 단락이 이루어지듯이 단락은 다시 장(chapter), 절(section), 소절(subsection)과 같은 상위 수준의 단위들로 논리에 맞추어 구조적으로 배치하여야 한다. 이러한 상위 수준의 구조는 조판상으로는 자명하다. 예를 들어, `\section{The Structure of Text and Language}`라고 입력하는 것은 ‘절 나눔’을 하는 것이라는 것이 분명히 드러난다.

제 2 절 줄바꿈과 쪽나눔

2.1 단락 정렬

단행본은 각 행의 길이가 똑같도록 조판하는 경우가 많다. L^AT_EX은 단락 전체의 내용을 최적화하여 낱말 사이에 줄바꿈과 공백을 삽입한다. 필요하다면, 줄 끝에 잘 들어가지 않는 낱말을 분절하여 하이픈을 삽입하기도 한다. 단락은 문서 클래스에 정의된 대로 조판된다. 보통 단락 첫 줄은 들여쓰기를 하고 단락과 단락 사이에는 여분의 간격을 두지 않는다. 이에 관해서는 116 쪽의 제 6 장 3.2절에서 좀 더 자세히 다룬다.

어떤 경우에는 L^AT_EX으로 하여금 강제로 줄을 바꾸도록 해야 할 필요가 있다. 다음 명령은 새 단락을 시작하지 않으면서 줄을 바꾼다.

```
\\ 또는 \newline
```

다음 명령은 강제로 줄을 바꾸면서 쪽나눔은 일어나지 않도록 한다.

```
\\*
```

새 쪽을 시작하려면 다음 명령을 이용한다.

```
\newpage
```

다음 명령들도 줄바꿈과 쪽나눔에 사용된다.

```
\linebreak[n], \nolinebreak[n], \pagebreak[n], \nopagebreak[n]
```

이들 명령은 그 이름이 의미하는 대로 작동한다. 저자는 0에서 4까지의 숫자를 인수 n 에 지정해서 명령의 동작을 조정할 수 있다. n 을 4 미만으로 지정하면 L^AT_EX은 명령이 실행된 결과가 매우 좋지 않을 때 그 명령을 무시한다. “break” 명령과 “new” 명령을 혼동하면 안 된다. “break” 명령이 사용되어도 L^AT_EX은 여전히 쪽의 오른쪽 경계와 길이를 맞추려한다. 이에 관해서는 다음

절에서 설명한다. 정말로 “새 줄”을 시작하기를 원한다면 “새 줄”을 의미하는 명령을 사용하여야 한다. 명령의 이름이 그렇게 붙여진 이유가 있는 것이다.

L^AT_EX은 항상 최선의 줄바꿈을 하려고 한다. 만일 미리 정의된 수준에 걸맞게 줄바꿈을 할 수 없을 경우에는 그 줄이 오른쪽으로 빠져 나오게 된다. 이 때에 L^AT_EX은 입력 파일을 처리하는 과정에서 “overfull hbox”라는 경고를 표시한다. 이러한 현상은 L^AT_EX이 낱말을 적절히 분철할 수 없을 때에 가장 빈번히 발생한다.¹ `\sloppy` 명령을 이용하면 L^AT_EX의 줄바꿈 처리 수준을 약간 낮출 수 있다. 그렇게 하면 비록 최선의 최종 출력물을 얻지 못하더라도 낱말 간격을 늘려서 오른쪽으로 빠져나오는 줄을 없앤다. 이 경우에는 “underfull hbox”라는 경고가 표시된다. 그러나 대부분의 경우 생성된 결과물의 모양은 썩 좋지 않다. L^AT_EX을 원래의 작동 방식으로 되돌리려면 `\fussy` 명령을 사용한다.

2.2 분철

L^AT_EX은 필요하다면 언제나 낱말을 분철(hyphenate)한다. 분철 알고리즘이 적절한 분철 위치를 찾지 못하면 다음의 명령을 사용하여 T_EX에게 예외 처리를 지시하여 문제를 해결할 수 있다.

```
\hyphenation{word list}
```

이 명령은 인자에 나열된 낱말들이 “-”으로 표시된 곳에서만 분철되도록 한다. 이 명령의 인자로 사용한 낱말들은 L^AT_EX이 보통 글자로 간주하는 문자와 기호들로만 구성되어야 한다. 분철 힌트는 분철 명령이 주어질 때에 활성화되는 언어를 위해 저장한다. 즉, 분철 명령을 문서의 전처리부에 두면 영어의 분철에 영향을 미치게 된다. 분철 명령을 `\begin{document}` 다음에 두면서 `babel`과 같은 특정 언어 지원 패키지를 이용하면, 분철 힌트는 `babel`을 통해 활성화된 언어에서 작동할 것이다.

다음의 예는 “hyphenation” 또는 “Hyphenation”의 분철을 허용하고 “FORTRAN”, “Fortran” 그리고 “fortran”은 분철을 허용하지 않는다. 특수 문자나 기호는 인자에 쓰일 수 없다.²

예:

```
\hyphenation{FORTRAN Hy-phen-a-tion}
```

`\-` 명령은 낱말에 임의의 하이픈을 삽입한다. 또한 이 위치는 그 낱말에서 분철이 허용되는 유일한 위치가 된다. 이 명령은 특수 문자(예를 들어 강세 표

¹L^AT_EX은 이런 문제(overfull hbox)가 발행했을 때에 경고와 함께 해당 줄을 보여주는 하지만, 실제로 그 해당 줄을 찾는 것은 쉽지 않다. `\documentclass` 명령의 선택 사항으로 `draft`를 이용하면 이런 줄들이 있는 오른쪽 여백에 굵은 검정 선을 그려 준다.

²분철 명령을 내리는 단어와 단어 사이에 콤마(,) 등의 구분자가 없음에 주의하라. —[역주]

시 문자)를 포함하고 있는 낱말의 경우 특히 유용하다. L^AT_EX은 특수 문자가 포함된 낱말을 자동으로 분철하지 못하기 때문이다.³

I think this is: su\~per\~cal\~%
i\~frag\~i\~lis\~tic\~ex\~pi\~%
al\~i\~do\~cious

I think this is: supercalifragilisticexpialidocious

여러 낱말들을 줄바꿈 없이 같은 줄에 두고 싶을 때에는 다음과 같이 한다.

`\mbox{text}`

이렇게 하면 인자로 주어진 낱말들은 어떠한 경우에도 함께 묶여 다니게 된다.

My phone number will change soon.
It will be \mbox{0116 291 2319}.

The parameter
`\mbox{\emph{filename}}` should
contain the name of the file.

My phone number will change soon. It will
be 0116 291 2319.

The parameter *filename* should contain the
name of the file.

`\fbox`는 `\mbox`와 비슷한데, 내용 주위에 테두리를 추가로 그려 준다.

제 3 절 미리 정의된 문자열

앞의 몇몇 예에서 특별한 문자열을 식자하는 데 쓰이는 간단한 L^AT_EX 명령을 사용하였다.

명령어	사용예	설명
<code>\today</code>	2005년 11월 5일	현재 사용 언어에서의 현재 날짜 표기
<code>\TeX</code>	T _E X	최고의 조판 시스템의 이름
<code>\LaTeX</code>	L ^A T _E X	지금 우리가 배우고 있는 것의 이름
<code>\LaTeXe</code>	L ^A T _E X 2 _ε	L ^A T _E X의 최신판

제 4 절 특수 문자와 기호

4.1 따옴표

타자기에서처럼 "를 따옴표로 사용해서는 안 된다. 출판물에서는 특별한 여는 따옴표와 닫는 따옴표를 사용한다. L^AT_EX에서는 두 개의 ‘(grave accent)

³OT1 폰트 인코딩을 사용하는 CM 글꼴의 경우 이 말은 옳바르다. 그러나 babel 패키지와 함께 T1 인코딩의 글꼴을 쓰면 강제 표시 문자가 들어 있는 단어의 분철이 가능해진다. 이에 대한 것은 다국어 지원의 문제와 폰트 인코딩 문제에 대해 다루고 있는 이 책의 곳곳을 참조하라. 한편 한국어 표기는 분철이 없고 따라서 분철부호도 쓰지 않는 대신 몇 가지 예외를 제외하고 모든 글자 앞뒤에서 행나눔을 할 수 있다. -[역주]

와 두 개의 ’(vertical quote)를 각각 여는 따옴표와 닫는 따옴표로 사용한다. 작은 따옴표의 경우에는 각각의 기호를 한 번씩만 사용하면 된다.

“Please press the ‘x’ key.”

“Please press the ‘x’ key.”

위의 L^AT_EX 명령 입력 예가 어떠한 글꼴을 사용해도 제대로 된 모양을 보여 주지 못하는 것은 어쩔 수 없다. 그러나 분명 여는 따옴표로는 표준 자판 제일 윗줄의 제일 왼쪽 키에 할당되어 있는 문자(back-tick 혹은 grave accent)를, 닫는 따옴표로는 자판 가운데줄 오른쪽 끝에 있는 작은따옴표 문자(vertical quote)를 사용해야 한다.

4.2 대시와 하이픈

L^AT_EX에서는 네 종류의 대시를 사용할 수 있다. 세 종류의 대시는 연속 입력하는 대시의 개수가 각기 다르다. 네 번째 것은 사실 대시가 아니라 수학에서 쓰이는 빼기 부호이다.

daughter-in-law, X-rated\\
pages 13--67\\
yes---or no? \\
\$0\$, \$1\$ and \$-1\$

daughter-in-law, X-rated
pages 13-67
yes—or no?
0, 1 and −1

이 대시들은 각각 ‘,’ 하이픈, ‘,’ 엔대시, ‘—’ 엠대시 그리고 ‘−’ 빼기 부호라고 한다.⁴

4.3 틸데 (~)

웹 주소에서 흔히 볼 수 있는 문자가 틸데이다.⁵ L^AT_EX에서 이 문자를 표시하려면 \~를 쓸 수 있다. 그러나 그 결과 ~는 원하는 대로 표시되지 않을 것이다. 따라서 다음과 같은 방법을 쓰도록 한다.

http://www.rich.edu/~bush \\
http://www.clever.edu/\$\sim\$demo

http://www.rich.edu/~bush
http://www.clever.edu/~demo

인터넷 주소에서 틸데 문자를 그대로 사용하도록 하는 url 또는 hyperref 패키지의 \url 명령이 있다. hyperref 패키지를 쓰면 이 명령의 인자로 사용한

⁴한글 맞춤법에서는 ‘대시’를 ‘줄표’로 ‘하이픈’은 ‘붙임표’라고 부른다. 길이에 따른 대시의 종류 구분은 없으며 사용법에도 차이가 있다. —[역주]

⁵틸데 문자는 한글 맞춤법에서 규정하고 있는 물결표(~)과 다른 것이며 그 용법에도 차이가 있다.—[역주]

인터넷 주소에 하이퍼링크가 만들어진다. 그밖에 좋은 점은 인터넷 주소에 해당하는 부분이 타입라이터 글꼴로 식자되면서도 적당히 행나눔이 된다는 점이다.

```
\url{http://people.ktug.or.kr/~karnes/%  
demo/lshort-kr/index.html}
```

```
http://people.ktug.or.kr/~karnes/demo/  
lshort-kr/index.html
```

4.4 도 기호 (°)

L^AT_EX에서 도 기호를 표시하는 방법을 아래 예에서 볼 수 있다.

```
It's $-30\,\mathrm{C}^\circ$.  
I will soon start to  
super-conduct.
```

```
It's -30°C. I will soon start to super-  
conduct.
```

textcomp 패키지를 이용하면 도 기호를 `\textcelsius` 명령으로 표시할 수 있다.

4.5 유로 통화 기호 (€)

이제 돈에 관해서 글을 쓰려면 유로 기호가 필요한 시대가 되었다. 최근의 많은 글꼴들이 유로 기호를 포함하고 있다. 유로 기호를 식자하려면 우선 다음과 같이 문서의 전처리부에 textcomp 패키지를 불러들인다.

```
\usepackage{textcomp}
```

그 다음에는 다음 명령으로 유로 기호를 표시할 수 있다.

```
\texteuro
```

현재 사용하고 있는 글꼴에 유로 기호가 들어 있지 않거나 들어 있더라도 마음에 들지 않을 때에는 다음의 두 가지 방법을 고려해볼 수 있다.

첫번째 방법은 eurosym 패키지를 이용하는 것이다.

```
\usepackage[official]{eurosym}
```

이 패키지는 official 옵션으로 공식 유로 기호를 식자할 수 있게 한다. 공식 유로 기호 대신 현재 사용 중인 글꼴에 어울리는 유로 기호를 쓰려 한다면 official 대신 gen을 패키지 선택 사항으로 쓰면 된다.

어도비 유로 폰트가 자신의 시스템에 설치되어 있다면(이 글꼴은 [ftp://ftp.adobe.com/pub/adobe/type/win/all](http://ftp.adobe.com/pub/adobe/type/win/all)에서 무료로 구할 수 있다.) europs

패키지의 `\EUR` 명령을 이용하여 현재 사용하는 글꼴과 어울리는 유로 기호를 식자할 수 있다.

많은 기호 문자를 제공하는 `marvosym` 패키지에도 유로 기호가 들어 있고 `\EUR` 명령으로 표시할 수 있다. 이 패키지의 단점은 기울인꼴 및 굵은꼴 유로 기호를 지원하지 않는다는 것이다.

표 2.1: 여러 가지 유로 기호들

package	command	roman	sans-serif	typewriter
eurosym	<code>\euro</code>	€	€	€
[gen]eurosym	<code>\euro</code>	€	€	€
europs	<code>\EUR</code>	€	€	€
marvosym	<code>\EUR</code>	€	€	€

4.6 줄임표 (...)

타자기에서는 쉼표나 마침표가 같은 넓이를 차지한다. 그러나 책을 인쇄할 때에는 이 기호들은 아주 적은 공간만 차지하며 앞선 글자에 매우 가깝게 붙어서 식자된다. 따라서 줄임표를 그냥 세 개의 마침표를 찍어서 표시하면 자간이 잘못되어 제대로 표시되지 않는다. 그러므로 세 개의 마침표를 찍는 대신에 다음과 같이 특별한 명령을 사용해야 한다.⁶

```
\ldots
```

Not like this ... but like this:\\
New York, Tokyo, Budapest, \ldots

Not like this ... but like this:
New York, Tokyo, Budapest, ...

4.7 합자

어떤 문자들의 조합은 한 글자씩을 따로따로 이어서 식자하지 않고 특별한 기호를 이용하여 식자한다. 이것을 합자(ligature)라고 한다.

ff fi fl ffi... instead of ff fi fl ffi ...

⁶한글 맞춤법에서 규정하고 있는 ‘줄임표’는 여섯 개의 가운데점들을 이용하여 표시한다. —[역주]

이 합자 기능을 이용하지 않으려면 `\mbox{}`를 합쳐지는 두 글자 사이에 끼워 넣으면 된다. 두 낱말이 합쳐져서 이루어진 낱말의 경우 이렇게 할 필요가 있다.⁷

```
\Large Not shelfful\\
but shelf\mbox{ }ful
```

Not shelfful
but shelfful

4.8 강세 부호와 특수 문자

L^AT_EX에서는 여러 언어에서 사용되는 강세 부호와 특수 문자를 쓸 수 있다. 표 2.2에서 영문자 o에 적용된 온갖 종류의 강세 부호를 볼 수 있다. 물론 강세 부호는 다른 문자에도 적용할 수 있다.

i와 j 위에 강세 부호를 붙이려면 이 두 문자 위의 점을 없애야 하는데, `\i`, `\j`라고 입력하면 된다.

```
H\^otel, na\"i ve, \'el\'eve,\\
sm\o rrebr\o d, !'Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

표 2.2: 강세 부호와 특수 문자

ò	\‘o	ó	\’o	ô	\^o	õ	\~o
ō	\=o	ô	\.o	ö	\"o	ç	\c c
ö	\u o	ö	\v o	ő	\H o	q	\c o
q	\d o	q	\b o	ôo	\t oo		
œ	\oe	Œ	\OE	æ	\ae	Æ	\AE
å	\aa	Å	\AA				
ø	\o	Ø	\O	ı	\l	Ł	\L
ı	\i	ı	\j	ı	!‘	ı	?‘

⁷합자 기능을 지원하느냐 하는 것은 폰트 디자인에도 관련되는 문제이다. 그러므로 어떤 영문 폰트를 쓰느냐에 따라 지원되는 합자의 범위가 달라질 수 있다. Computer Modern Roman 글꼴은 기본적인 합자 기능을 대부분 지원한다. —[역자]

제 5 절 다국어 지원

영어 이외의 다른 언어로 된 문서를 작성할 때에는 \LaTeX 은 다음 세 가지 영역이 제대로 설정되어야 한다.

1. 모든 자동 생성 문자열⁸을 새로운 언어에 맞게 바꾸어야 한다. Johannes Braams의 `babel` 패키지를 이용하면 이 설정을 여러 언어에 대해 쉽게 할 수 있다.
2. 새로운 언어의 분철 규칙을 \LaTeX 에게 알려 주어야 한다. 새로운 분철 규칙을 \LaTeX 으로 불러들이는 것은 아주 간단하지는 않다. 즉, 새로운 분철 패턴이 작동하도록 포맷 파일을 새로 생성하여야 한다. 이에 관해서는 *Local Guide* [5]에 자세히 설명되어 있을 것이다.
3. 각 언어마다 다른 조판 규칙. 예를 들어, 프랑스어에서는 쌍점 문자(:) 다음에 반드시 공백을 하나 넣어야 한다.

사용하는 시스템이 제대로 설정되어 있다면 `\documentclass` 명령 다음에 아래 명령을 두어서 `babel` 패키지를 불러 올 수 있다.

```
\usepackage[language]{babel}
```

사용 중인 \LaTeX 시스템에서 이용할 수 있는 언어의 목록이 컴파일러가 작동할 때마다 표시될 것이다. `babel`은 사용자가 선택한 언어에 알맞은 분철 규칙을 자동으로 활성화할 것이다. 만일 사용 중인 \LaTeX 포맷이 선택한 언어의 분철을 지원하지 않는다면, `babel`은 분철 기능을 중지한 상태로 작동할 것이다. 이는 조판된 문서의 외양에 좋지 않는 영향을 미친다.

`babel`은 몇 언어에 대해서 특수 문자 입력을 쉽게 할 수 있는 새로운 명령어를 지원한다. 예를 들어 독일어에서는 여러 종류의 움라우트(äöü) 문자가 쓰인다. `babel` 패키지를 이용하면 `\"o`라고 하는 대신에 `o`라고 하는 것만으로 ö를 입력할 수 있다.

`babel`을 다음과 같이 여러 언어와 함께 불러온다면

```
\usepackage[languageA,languageB]{babel}
```

다음의 명령을 사용하여 현재 언어를 지정해 주어야 한다.

```
\selectlanguage{languageA}
```

대부분의 최근 컴퓨터 시스템에서는 키보드로 특수 문자를 직접 입력할 수

⁸차례, 그림 차례 등.

있다. 다양한 언어와 컴퓨터 플랫폼에서 사용되는 여러 종류의 입력 인코딩을 다루기 위해서 \LaTeX 에서는 `inputenc` 패키지를 이용한다.

```
\usepackage[encoding]{inputenc}
```

이 패키지를 사용할 때에는 다른 인코딩을 사용하는 컴퓨터를 사용하는 사람은 작성된 파일을 표시할 수 없을 수도 있다는 점을 염두에 두어야 한다. 예를 들어, 독일어 움라우트 ä는 OS/2에서 132로 인코딩되고, ISO-LATIN 1을 사용하는 유닉스 시스템에서는 228로 인코딩되지만, 윈도우의 cp1251 Cyrillic 인코딩에는 이 글자가 아예 없다. 그러므로 이 기능은 주의해서 사용해야 한다. 아래 표에 보인 인코딩 방식들은 사용자의 시스템 유형에 따라 유용할 수 있다.⁹

운영 체제	인코딩	
	western Latin	Cyrillic
Mac	applemac	macukr
Unix	latin1	koi8-ru
Windows	ansinew	cp1251
DOS, OS/2	cp850	cp866nav

인코딩이 서로 충돌하여 한 문서 안에 함께 존재하기 힘든 언어들이 포함된 다국어 문서를 작성해야 한다면 `ucs` 패키지를 이용해서 유니코드를 사용하는 것이 나을 수도 있다.

```
\usepackage{ucs}
\usepackage[utf8]{inputenc}
```

위 명령을 통해서 한 글자가 적게는 한 바이트에서 많게는 네 바이트까지로 인코딩되는 멀티바이트 인코딩인 `utf8`을 \LaTeX 입력 파일에서 사용할 수 있다.

글꼴 인코딩은 또 다른 문제로 어떤 \TeX 글꼴의 어느 위치에 각각의 문자가 저장되는가를 정의하는 것이다. 여러 개의 입력 인코딩이 하나의 글꼴 인코딩에 대응되면 요구되는 글꼴 집합의 개수를 줄일 수 있다. 글꼴 인코딩은 `fontenc` 패키지가 담당한다.

```
\usepackage[encoding]{fontenc}
```

여기서 `encoding`은 글꼴 인코딩이다. 동시에 여러 개의 인코딩을 불러들이는 것도 가능하다.

⁹라틴 및 키릴 기반 언어들의 입력 인코딩 방식에 관해서는 `inputenc.dtx`와 `cyinpenc.dtx` 패키지에 따라오는 문서를 참조하라. 패키지 문서를 만드는 방법은 77 쪽의 제 6장에서 설명한다.

L^AT_EX에서 사용하는 기본 글꼴 인코딩은 OT1으로 오리지널 컴퓨터 모던 (Computer Modern) T_EX 글꼴을 인코딩한 것이다. 이 인코딩은 7 비트 아스키 문자 집합을 위한 128 개의 문자만을 포함하고 있다. 강세 표시된 문자가 붙은 문자가 필요할 경우 T_EX은 일반 문자와 강세 부호를 조합하여 강세 표시 문자를 생성한다. 출력물의 모양새는 완벽하지만 이 방식은 강세 표시된 문자를 포함한 낱말의 분철을 불가능하게 만들어 버린다. 더구나 이 방식으로는 그리스 문자나 키릴 문자와 같은 비라틴 알파벳 문자는 말할 것도 없고, 라틴 문자에 포함된 몇몇 문자도 보통 문자와 강세 부호를 조합해서 만들어 낼 수 없다.

이러한 단점을 극복하기 위해서 몇몇 8 비트 CM 계열 글꼴이 만들어졌다. T1 인코딩의 *Extended Cork*(EC) 글꼴은 라틴 표기법에 기반한 대부분의 유럽어를 위한 문자와 문장 부호를 포함하고 있다. LH 글꼴은 키릴 문자를 이용하는 언어로 된 문서를 조판하기 위해 필요한 문자들을 포함하고 있다. 키릴 문자는 문자 개수가 많기 때문에 T2A, T2B, T2C, 그리고 X2의 네 개의 글꼴 인코딩에 나누어 배열되어 있다.¹⁰ CB 묶음은 그리스어 문서 작성을 위한 LGR 인코딩의 글꼴을 포함하고 있다.

이 글꼴들을 이용해서 비영어 문서의 분철을 강화 혹은 작동시킬 수 있다. CM 계열 글꼴을 이용하는 또 다른 장점은 이 글꼴이 모든 굵기와 모양, 그리고 다양한 크기의 CM 글꼴을 지원한다는 것이다.

5.1 한국어 지원¹¹

L^AT_EX에서 한국어¹²로 작성된 문서를 조판하려면 다음의 패키지들 가운데 하나를 이용할 수 있다.

- H^AL_TE_X

은광희 님이 제작한 패키지로 L^AT_EX에서 EUC-KR¹³로 인코딩된 한글을 식자하고 한글 서식의 문서를 작성할 수 있다.¹⁴ 오랫동안 한글 L^AT_EX

¹⁰각각의 인코딩이 지원하는 언어의 목록은 [11]에서 볼 수 있다.

¹¹이 절은 원래 한국어 번역본 3.20판에 김강수가 쓴 글로 처음 포함되었다. 그 후 신정식 님의 번역으로 lshort 영문판에도 실리게 되었다. 4.17판을 번역하면서 지난 몇 년 동안에 달라진 한글 환경에 대한 내용을 추가하여 갱신하였다. 번역자 중 이기황이 초고를 집필하였다.

¹²특정 문맥에서는 혼용되기도 하지만, 엄밀히 구분하면 ‘한국어’는 영어, 일본어 등과 대비되는 ‘언어’를 일컫는 말이며, ‘한글’은 ‘알파벳’, ‘히라카나’ 등과 대비되는 ‘문자 체계’를 일컫는 말이다.

¹³한글 문서의 입력 인코딩으로 KS X 1001에서 규정한 이른바 KS 완성형을 사용하는 문자 집합을 EUC-KR이라 부르기로 하겠다. MS Windows 운영체제는 EUC-KR 문자를 기초로 이를 확장한 CP949(윈도 확장 완성형)를 사용자 인터페이스에서 사용하고 있으므로 EUC-KR 범위의 문자는 CP949와 코드가 겹친다. 그러나 CP949에서 표현할 수 있는 “모든 한글”을 모두 표현할 수 없고 한글의 경우 그 표현 범위가 2350자로 제한된다. 이와 같은 한글 표현의 제한을 넘어서는 방법은 유니코드를 사용하는 것이다. 리눅스나 매킨토시 운영체제에서는 점차 유니코드/UTF-8을 시스템의 기본 문자집합으로 채택하는 예가 증가하고 있어서 이와 관련된 불편이 보고되고 있다. 아무튼 H^AL_TE_X을 사용하고 이것을 L^AT_EX으로 처리하려 하는 한 한글 입력 문자의 인코딩은 EUC-KR이어야 한다. dhucs 등은 이러한 제한을 극복하기 위해 제작된 것이다.

¹⁴H^AL_TE_X은 <http://project.ktug.or.kr/hlatex/>에서 구할 수 있다.

의 사실상 표준으로 쓰였으며 지금도 가장 많은 사용자를 가지고 있는 패키지이다. UHC 글꼴¹⁵을 기본 글꼴로 사용한다.

- dhucs

김도현 님과 김강수 님에 의해 비교적 최근에 개발된 패키지로 UTF-8 유니코드로 입력된 한글을 L^AT_EX 문서에서 식자하고 조판할 수 있다.¹⁶ 이 패키지는 27 쪽에서 잠깐 언급한 Dominique Unruh 씨의 unicode 패키지(ucs 패키지)가 제공하는 UTF-8 처리 기능을 한글 식자에 이용한다. 이 패키지는 H^AL^AS^EX을 사용할 때 만날 수 있는 몇 가지 패키지 충돌을 완화할 수 있으며, PDF 출력을 얻는 데에 여러 가지로 유리한 점 등이 있어 사용자가 점차 늘고 있다. dhucs 패키지는 유니코드 기반으로 동작하기 때문에 일본어와 중국어와 같은 다국어 식자도 부분적으로 가능하며, 다른 언어 지원도 필요하다면 구현 가능하다.

- 그밖에 차재춘 님이 만든 hL^AT_EXp, Werner Lemberg 씨가 유지하는 CJK 패키지도 한국어 조판에 이용할 수 있다.¹⁷ hL^AT_EXp는 EUC-KR로 인코딩된 한글을 처리하는 시스템이라는 점에서 H^AL^AS^EX과 비슷하나 사용하는 글꼴은 크게 차이가 난다. CJK 패키지는 EUC-KR 한글 이외에 UTF-8 유니코드 입력 파일도 처리할 수 있으며, 일본어/중국어/한국어 및 (4.5.1 판부터는) 타이어도 쓸 수 있다.¹⁸

위 패키지들 가운데 H^AL^AS^EX, dhucs, hL^AT_EXp는 모든 자동 생성 텍스트를 한국어로 (선택적으로) 번역해 주며, 한국어 문서를 위한 다양한 조판 양식을 지원한다. 또한 자동 조사 처리 기능도 제공한다.

이들 패키지들 가운데 하나를 사용하여 한글 문서를 작성하기 위해서는 다음 사항들이 고려되어야 한다.

1. 한글 입력 파일을 만들 수 있는 환경이 필요하다. 한글 입력 파일은 plain text이지만 ASCII 범위 밖의 문자를 사용한다. 따라서 사용하는 한글 식자용 패키지에 따라 EUC-KR 혹은 UTF-8로 인코딩된 한글 텍스트를 입력하고 저장할 수 있는 편집기와 글꼴 등의 한글 환경이 마련되어야 한다. 세부적인 사항은 사용자가 사용하는 시스템의 하드웨어 및 소프트웨어 플랫폼에 따라 달라진다.

¹⁵이 글꼴은 은광희 님께서 직접 제작한 PostScript type1 글꼴이다. 원래 명칭은 통합 한글 코드(Unified Hangul Code)를 모두 표현할 수 있는 글꼴이라는 뜻으로 붙여진 명칭이었으나 현재는 오히려 “은광희의 한글 조합 폰트(Un’s Hangul Composite Fonts)”라는 뜻으로 사용자들에게 받아들여지고 있다. 이 글꼴은 트루타입 “은글꼴”의 바탕이 되었다.

¹⁶dhucs 패키지는 <http://faq.ktug.or.kr/faq/DHUCs/>에서 구할 수 있다.

¹⁷이들 패키지는 <http://knot.kaist.ac.kr/htex/>와 CTAN:/tex-archive/language/korean/CJK/에서 각각 찾아볼 수 있다.

¹⁸CJK 패키지의 다양한 응용에 관해서는 <http://faq.ktug.or.kr/faq/CJKLaTeX/>을 참조하라.

2. 한글 출력물을 만들 수 있는 글꼴이 있어야 한다. 위의 한국어 패키지들은 각각 몇 가지 한글 글꼴을 함께 제공한다. H_AT_EX은 패키지 저자인 은광희 님이 제작한 type 1 형식의 UHC 글꼴이 함께 배포되고 있다. dhucs 패키지는 UHC 글꼴을 트루타입(truetype)으로 이식한 은글꼴을 기본 글꼴로 채택하고 있다. 최근 다양한 한글 트루타입 글꼴을 비교적 쉽게 L^AT_EX에서 사용할 수 있는 방법이 개발되어, 약간의 제한 사항이 있기는 하지만 L^AT_EX에서의 한국어 조판을 위한 글꼴 선택의 폭은 매우 넓어졌다.

지금 읽고 있는 책자를 조판하는데 사용한 dhucs 패키지를 이용하여 한국어 문서를 작성할 때에는 전처리부에서 다음과 같이 선언한다.¹⁹

```
\usepackage[hangul,nonfrench]{dhucs}
```

dhucs가 제공하는 패키지 선택 사항은 다음과 같다.

hangul 한글 문서 서식이 설정되어 자동 생성 문자열이 모두 한국어로 바뀐다. 기본값은 한글 문서 서식을 설정하지 않는 것이다.

nojosa 자동 조사 기능을 끈다. beamer 등의 패키지와 함께 쓸 때 유용하다.

hanja 한글 문서 서식을 설정하는 것은 hangul 선택 사항과 같으나, 한글 이름 대신 한자 이름을 사용한다.

nonfrench 한글 문장에 대해서도 nonfrench-spacing을 적용한다. 즉, 마침표 등의 뒤에 나오는 공백이 다른 공백보다 조금 더 넓다. 기본값은 french-spacing을 적용하는 것이다.

dhucs와 함께 제공되는 부가 패키지들은 다음과 같다.

dhucs-interword 한국어 문서의 자간과 낱말 간격을 설정한다. 선택 사항으로는 default와 HWP를 쓸 수 있다.

dhucs-setspace 한국어 문서의 줄간격을 설정한다. 이 패키지를 이용하려면 hangul 선택 사항을 반드시 지정해야 한다.

dhucs-gremph \itshape에 해당하는 한글 문자를 그래픽 글꼴로 바꾼다. 그래픽 대신 굵은 글꼴로 대신하는 bfemph 선택 사항을 쓸 수도 있다.

패키지의 설치 방법, 기본 글꼴의 변경, PDF 출력을 얻는 방법 등 dhucs의 자세한 사용법은 앞서 언급한 dhucs의 홈페이지를 참조하기 바란다. dhucs 이외의 다른 패키지를 이용한 L^AT_EX에서의 한국어 조판과 이와 관련된 사항에 관한 더 자세한 정보는 KTUG(Korean T_EX User Group)에서 운영하는 <http://www.ktug.or.kr>과 <http://faq.ktug.or.kr/mywiki>에서 얻을 수 있다.

¹⁹한국어 문서의 입력 인코딩이 UTF-8이어야 한다는 점을 잊지 않도록 하자.

표 2.3: 포르투갈어 문서를 위한 전처리부

```
\usepackage[portuguese]{babel}
\usepackage[latin1]{inputenc}
\usepackage[T1]{fontenc}
```

5.2 포르투갈어 지원

By Demerson Andre Polli <polli@linux.ime.usp.br>

포르투갈어를 위한 분철 규칙을 사용하면서 모든 자동 생성 문자열을 이 언어로 바꾸려면 다음 명령을 이용한다.

```
\usepackage[portuguese]{babel}
```

브라질에서는 언어를 `brazilian`로 바꾼다.

포르투갈어에서는 강세 표시 문자를 많이 사용한다. 강세 표시 문자들을 일일이 명령형으로 바꾸지 않고 입력된 그대로 처리하려면 다음과 같이 지시한다.

```
\usepackage[latin1]{inputenc}
```

또한 분철이 제대로 작동하게 하려면 다음과 같이 한다.

```
\usepackage[T1]{fontenc}
```

포르투갈어 문서를 위한 전처리부의 내용은 표 2.3에 나와 있다. 여기서 Mac이나 DOS에서는 쓸 수 없는 `latin1` 입력 인코딩을 이용하고 있는 것에 주의하라. 사용 시스템에 적합한 인코딩을 사용해야 한다.

5.3 프랑스어 지원

By Daniel Flipo <daniel.flipo@univ-lille1.fr>

이 소절에서는 \LaTeX 를 이용하여 프랑스어 문서를 작성하는데 필요한 몇 가지 힌트를 소개한다. 프랑스어 지원 기능은 다음의 명령으로 불러들일 수 있다.

```
\usepackage[frenchb]{babel}
```

역사적인 이유로 인해 프랑스어를 위한 `babel`의 선택 사항으로 `french`가 아닌 `frenchb`나 `francais`가 사용되는 것을 주의하라.

사용 중인 L^AT_EX 시스템이 제대로 설정되었다면 위의 명령은 프랑스어 분철을 작동시킨다. 또한 모든 자동 생성 문자열이 프랑스어로 바뀐다. 예를 들어 `\chapter`는 `Chapitre`로 바뀌고, `\today`는 프랑스어 형식으로 현재 날짜가 표시된다. 프랑스어 입력 파일을 쉽게 작성할 수 있도록 몇 개의 새 명령도 쓸 수 있게 된다. 표 2.4를 보라.

표 2.4: 프랑스어를 위한 특수 명령어

<code>\og guillemets \fg{}</code>	« guillemets »
<code>M\up{me}, D\up{r}</code>	M ^{me} , D ^r
<code>1\ier{ }, 1\iere{ }, 1\ieres{ }</code>	1 ^{er} , 1 ^{re} , 1 ^{res}
<code>2\ieme{ } 4\iemes{ }</code>	2 ^e 4 ^{es}
<code>\No 1, \no 2</code>	N ^o 1, n ^o 2
<code>20~\degres C, 45\degres</code>	20 °C, 45°
<code>\bsc{M. Durand}</code>	M. DURAND
<code>\nombre{1234,56789}</code>	1 234,567 89

프랑스어를 사용할 때에는 나열 항목의 모양새 또한 바뀐다. `babel` 패키지의 `frenchb` 선택 사항의 기능, 사용법에 관해서는 `frenchb.dtx` 파일에 대해서 L^AT_EX을 실행에서 생기는 `frenchb.dvi`를 읽어보라.

5.4 독일어 지원

이 소절에서는 L^AT_EX을 이용하여 독일어 문서를 작성하는 방법에 대해 간략히 소개한다. 독일어 지원은 다음과 같은 명령으로 읽어들이 수 있다.

```
\usepackage[german]{babel}
```

위와 같이 하면, L^AT_EX 시스템이 제대로 설정되었을 경우 독일어 분철 기능이 작동한다. 또한 모든 자동 생성 텍스트들이 독일어로 바뀐다. 예를 들어, “Chapter”는 “Kapitel”로 바뀐다. 그리고 `inputenc` 패키지를 사용하지 않더라도 독일어 입력 파일을 쉽게 작성할 수 있는 새 명령들도 사용할 수 있게 된다. 표 2.5를 보라. `Inputenc` 패키지를 사용하면 이 명령들은 필요가 없다. 그러나 입력 문서는 특정 인코딩이 지원되는 환경에서밖에 쓸 수 없게 된다.

독일어로 된 책에서 종종 프랑스식 따옴표(«guillemets»)를 찾아볼 수 있다. 그러나 독일식 조판에서는 이들을 다르게 사용한다. 독일어 책에서의 따옴표는 »이렇게« 조판된다. 스위스 독일어에서는 프랑스어에서와 마찬가지로 »이렇게« 조판한다.

표 2.5: 독일어 특수 문자들

"a	ä	"s	ß
"‘	”	"’	“
"< or \flqq	«	"> or \frqq	»
\flq	<	\frq	>
\dq	"		

큰 문제는 \flq와 같은 명령을 사용할 때에 발생한다. 기본 글꼴인 OT1 글꼴을 사용하면 프랑스식 따옴표는 수학 기호 “<<”처럼 보인다. 프랑스식 따옴표를 사용하려면 필요한 기호를 포함하고 있는 T1 인코딩 글꼴을 이용해야 한다.(\usepackage[T1]{fontenc})

5.5 키릴 문자 지원

By Maksym Polyakov <polyama@myrealbox.com>

babel 3.7h판은 키릴 문자를 쓰는 불가리아어, 러시아어, 그리고 우크라이나어 조판을 위한 T2* 인코딩을 지원한다.

키릴 문자 지원은 표준 L^AT_EX에 기반하여 fontenc와 inputenc 패키지를 이용하여 이루어진다. 수학 모드에서 키릴 문자를 쓰려면 다음과 같이 mathtext 패키지를 fontenc 패키지 전에 불러야 한다.²⁰

```
\usepackage{mathtext}
\usepackage[T1,T2A]{fontenc}
\usepackage[koi8-ru]{inputenc}
\usepackage[english,bulgarian,russian,ukranian]{babel}
```

일반적으로 위의 세 언어를 위한 글꼴 인코딩은 babel 패키지가 알아서 기본 글꼴 인코딩을 선택한다. 그러나 한 문서에서 한 종류의 글꼴 인코딩만 사용할 수 있는 것은 아니다. 키릴 문자와 라틴 문자 언어가 함께 포함된 다국어 문서에서는 라틴 글꼴 인코딩을 명시적으로 포함시키는 것이 좋다. 문서 안에서 사용 언어가 바뀔 때에는 babel 패키지가 자동으로 적절한 글꼴 인코딩을 선택해 줄 것이다.

babel 패키지는, 분철 기능 작동, 자동 생성 문자열 번역, 개별 언어 의존적 조판 규칙(\frenchspacing 등)을 활성화하는 것 이외에도 불가리아어, 러시아어, 우크라이나어의 표준에 맞는 조판을 위한 명령들을 쓸 수 있게 한다.

²⁰ $\mathcal{A}\mathcal{M}\mathcal{S}$ -L^AT_EX 계열 패키지들도 fontenc와 babel 패키지 전에 불러야 한다.

이 세 언어 각각에 대한 문장 부호도 지원한다. 키릴 대시(라틴 대시보다 약간 짧고 양쪽에 아주 작은 공백이 있음), 직접 인용, 인용 그리고 분철을 위한 명령들을 표 2.6에서 살펴볼 수 있다.

표 2.6: babel 패키지에 불가리아, 러시아, 우크라이나어 옵션을 주었을 때의 추가 정의

"	disable ligature at this position.
"-	an explicit hyphen sign, allowing hyphenation in the rest of the word.
"---	Cyrillic emdash in plain text.
"--~	Cyrillic emdash in compound names (surnames).
"--*	Cyrillic emdash for denoting direct speech.
"	like "-", but producing no hyphen sign (for compound words with hyphen, e.g. x-"y or some other signs as "disable/enable").
"~	for a compound word mark without a breakpoint.
"=	for a compound word mark with a breakpoint, allowing hyphenation in the composing words.
",	thinspace for initials with a breakpoint in following surname.
"‘	for German left double quotes (looks like „).
"’	for German right double quotes (looks like “).
"<	for French left double quotes (looks like <<).
">	for French right double quotes (looks like >>).

babel 패키지에 러시아어, 우크라이나어 옵션을 지정하면 \Asbuk과 \asbuk 명령이 정의된다. 이 명령들은 각각 \Alph와 \alph 명령처럼 숫자 표현 형식을 알파벳으로 바꾸어서 러시아어와 우크라이나어(둘 중 현재 문서에 활성화된 언어)의 대문자와 소문자로 찍는다. 불가리아어 옵션으로 사용되면 \enumBul과 \enumLat (\enumEng) 명령이 정의된다. \enumBul 명령을 쓰면 \Alph나 \alph와 같은 알파벳 숫자 형식을 불가리아어 자모로 찍을 수 있게 한다. 반면 \enumEng는 보통의 라틴 알파벳 숫자 형식을 그대로 쓸 수 있게 해준다. 기본 동작은 불가리아어 문자를 찍는것이다.

제 6 절 띄어쓰기 간격

출력물의 오른쪽 끝을 가지런히 정렬하기 위해서 L^AT_EX은 단어 사이의 간격을 적절히 조정한다. 글을 읽기 쉽게 하기 위해서 각 문장의 끝에는 공백이 약간 더 삽입된다. L^AT_EX은 문장이 마침표, 물음표 혹은 느낌표로 끝나는 것으로 가정한다. 대문자 다음에 바로 마침표가 오는 것은 약어일 경우가 많으므로 문장의 끝으로 여기지 않는다.

위의 가정에서 벗어나는 예외들은 사용자가 직접 지정해야 한다. 공백 문자 앞에 역사선 문자를 두면 폭이 늘어나지 않는 일정한 폭의 공백이 만들어

진다. 틸데 문자 ‘~’는 줄바꿈이 일어나지 않는 폭이 고정된 공백을 만든다. 마침표 앞에 \@ 명령을 쓰면 마침표 앞에 대문자가 있더라도 문장의 끝으로 처리한다.

```
Mr.~Smith was happy to see her\\
cf.~Fig.~5\\
I like BASIC\@. What about you?
```

```
Mr. Smith was happy to see her
cf. Fig. 5
I like BASIC. What about you?
```

마침표 다음에 오는 여분의 공백은 다음 명령으로 생기지 않게 할 수 있다.

```
\frenchspacing
```

이 명령이 쓰이면 L^AT_EX은 마침표 다음에 보통 문자 다음에 삽입하는 공백보다 더 넓은 공백을 삽입하지 않는다. 이러한 조판 방식은 참고 문헌의 경우를 제외하면 비영어권 언어에서 흔히 볼 수 있다. \frenchspacing을 쓰면 \@ 명령은 쓸 필요가 없다.

제 7 절 제목과 장, 절

독자가 글을 쉽게 읽어 내려가게 하려면, 글을 장(chapter), 절(section), 그리고 소절(subsection) 등으로 나누어야 한다. L^AT_EX에서는 장절 구분을 할 때에 장절 제목을 인자로 취하는 특별한 명령을 이용한다. 장절 구분 명령을 순서에 맞게 제대로 쓰는 것은 사용자의 몫이다.

article 클래스에서 사용할 수 있는 장절 명령은 다음과 같다.

```
\section{...}
\subsection{...}
\subsubsection{...}
\paragraph{...}
\subparagraph{...}
```

장절 번호 매김에 영향을 주지 않으면서 문서를 몇 개의 부(part)로 나누려면 다음 명령을 사용한다.

```
\part{...}
```

report와 book 클래스에서는 다음의 최상위 장절 명령을 추가로 쓸 수 있

다.

```
\chapter{...}
```

`article` 클래스에는 ‘장’이 없으므로, `article` 클래스로 작성된 문서는 책의 한 장으로 쉽게 삽입할 수 있다. 절간의 간격, 제목의 번호 매김과 글자 크기는 \LaTeX 이 자동으로 처리해준다.

다음 두 장절 명령은 약간 특별하다.

- `\part` 명령은 장의 번호 매김에 영향을 끼치지 않는다.
- `\appendix` 명령은 인자를 취하지 않으며, 장의 번호 매김을 숫자에서 문자로 바꾼다.²¹

\LaTeX 은 문서를 컴파일하면서 장절 제목과 쪽 번호를 기록해둔다. 차례는 문서의 직전 컴파일에서 기록해둔 장절 제목과 쪽 번호를 취해서 만들어낸다. 다음 명령은 이 명령이 위치한 곳에 차례를 만들어 넣는다.

```
\tableofcontents
```

새로 작성된 문서의 올바른 차례를 얻으려면 그 문서를 두 번 컴파일해야 한다. 어떠한 경우에는 세 번 컴파일하여야 한다. 그럴 경우 \LaTeX 이 컴파일이 더 필요하다는 것을 알려준다.

위에서 설명한 명령들은 모두 대응하는 “별표 붙은” 명령들이 있다. “별표 붙은” 명령은 보통의 명령 뒤에 *를 붙인 명령으로서 번호가 매겨지지 않으며 차례에도 나오지 않는 장절 제목을 만들어낸다. 예를 들어, `\section{Help}`에 별표를 붙이면 `\section*{Help}`가 된다.

보통 장절 제목은 본문에 입력된 그대로 차례에 나온다. 그러나 장절 제목이 너무 길어서 차례에 잘 맞지 않을 때도 있다. 그럴 때에는 실제 제목의 앞에 선택 인자로 차례에 나타날 짧은 제목을 지정하면 된다.

```
\chapter[Title for the table of contents]{A long
and especially boring title, shown in the text}
```

문서 전체의 표제는 다음의 명령으로 붙인다.

```
\maketitle
```

표제에 포함될 내용은 다음의 명령을 이용하여 지정해야 한다.

```
\title{...}, \author{...} 그리고 필요할 경우 \date{...}
```

이 명령들은 `\maketitle` 명령 앞에 놓아야 한다. `\author` 명령의 인자 안에

²¹`article` 클래스에서는 절 번호 매김을 바꾼다.

서 `\and` 명령으로 여러 명의 저자 이름을 구분하여 넣을 수 있다.

위에서 설명한 몇 가지 명령의 사용례는 9 쪽의 그림 1.2에 있다.

위에서 설명한 장절 명령 이외에 $\text{\LaTeX 2}_{\epsilon}$ 는 book 클래스에서 사용할 수 있는 명령 세 가지를 더 갖추고 있는데, 이들 명령은 출판물을 몇 부분으로 나눌 때에 유용하다. 이 명령들을 쓰면 보통 책에서 볼 수 있는 것처럼 각 부분별로 장절 제목 붙이기와 쪽 번호 매김을 다르게 할 수 있다.

`\frontmatter` 는 `\begin{document}` 바로 다음에 두어야 한다. 이 명령은 쪽 번호를 로마 숫자로 바꾸고 “별표 붙은” 명령을 쓴 것처럼 장절에 번호가 붙여지지 않게 한다.(예: `\chapter*{Preface}`) 그러나 장절 제목은 차례에 나타난다.

`\mainmatter` 는 책의 첫 장 바로 앞에 놓아야 한다. 이 명령은 쪽 번호를 아라비아 숫자로 바꾸고 쪽 번호 매김을 다시 시작하도록 한다.

`\appendix` 는 책의 부록 부분의 시작을 표시한다. 이 명령 이후에 나오는 장들은 숫자가 아닌 문자로 번호가 매겨진다.

`\backmatter` 는 참고 문헌, 찾아 보기와 같은 책의 마지막 부분 바로 앞에서 선언한다. 표준 문서 클래스에서는 이 명령의 외관상 효과는 없다.

제 8 절 상호 참조

단행본, 보고서, 논문에서는 그림, 표, 또는 본문의 특별한 부분을 상호 참조해야 하는 경우가 많다. \LaTeX 에서의 상호 참조는 다음 명령을 통해 이루어진다.

```
\label{marker}, \ref{marker} and \pageref{marker}
```

여기서 *marker*는 사용자가 지정하는 식별자이다. \LaTeX 은 `\ref` 명령을 해당하는 `\label` 명령이 쓰인 곳에 있는 절, 소절, 그림, 표, 정리 등의 번호로 바꾼다. `\pageref`는 해당하는 `\label` 명령이 나타난 쪽 번호를 표시한다.²² 장절 제목과 마찬가지로 번호는 직전 컴파일 결과에서 얻는다.

```
A reference to this subsection
\label{sec:this} looks like:
“see section~\ref{sec:this} on
page~\pageref{sec:this}.”
```

A reference to this subsection looks like: “see section 8 on page 37.”

²²이 명령들은 자신이 무엇을 참조하는지 모른다. `\label` 명령은 단지 이전에 생성된 번호를 저장하고 있을 뿐이다.

제 9 절 각주

각주는 다음의 명령으로 붙인다.

```
\footnote{footnote text}
```

각주는 현재 쪽의 하단에 인쇄된다. 각주는 항상 그 각주가 참조하는 낱말이나 문장 뒤에 놓아야²³ 한다. 그러므로 한 문장, 혹은 문장의 일부를 참조하는 각주는 쉼표나 마침표 뒤에 놓아야 한다.²⁴

```
Footnotes\footnote{This is
a footnote.} are often used
by people using \LaTeX.
```

Footnotes^a are often used by people using L^AT_EX.

^aThis is a footnote.

제 10 절 강조

글을 타자기로 작성할 때 중요한 낱말은 밑줄로 강조한다.

```
\underline{text}
```

그러나 인쇄물에서는 이탤릭으로 강조를 나타낸다.²⁶ L^AT_EX은 다음 명령으로 강조를 지시할 수 있다.

```
\emph{text}
```

이 명령의 실제 동작은 앞뒤 문맥에 따라 다르다.

```
\emph{If you use
emphasizing inside a piece
of emphasized text, then
\LaTeX{} uses the
\emph{normal} font for
emphasizing.}
```

If you use emphasizing inside a piece of emphasized text, then L^AT_EX uses the normal font for emphasizing.

²³예를 들어 ‘놓아야’라는 어절에 대한 각주는 여기에 와야 한다.

²⁴각주를 사용할 때 주의할 점이 있다. 각주를 만나면 독자의 관심이 본문에서 각주로 옮겨간다. 사람들은 당연히 호기심 많은 존재라서 결국 각주를 다 읽을 것이다. 그러니 차라리 말하고 싶은 바를 모두 본문에 넣자. 그러면 독자의 관심을 혼란스럽게 하지 않으면서도 전달하고 싶은 바를 다 전할 수 있으니 더 낫지 않겠는가?²⁵

²⁵원문의 이 부분에는 각주 안에 또다른 각주를 붙이는 재미있는 예를 설명없이 보여주고 있다. 이것은 \footnotemark와 \footnotetext{...}를 이용한 것으로 각주 안에 각주 표지를 찍기 위해서 전자를, 그리고 각주 내용을 표현하기 위해서 후자를 사용하였다. —[역주]

²⁶한글 맞춤법에서는 ‘드러냄표’를 이용한 강조에 관련된 규정이 있다. 이것은 강조하고자 하는 어구에 점(가로쓰기에서는 상점, 세로쓰기에서는 방점)을 찍는 것이다. —[역주]

L^AT_EX에게 무엇인가를 강조하게 하는 것과 다른 글꼴을 사용하게 하는 것은 다른 것이라는 것을 알아야 한다.

```
\textit{You can also
\emph{emphasize} text if
it is set in italics,}
\textsf{in a
\emph{sans-serif} font,}
\texttt{or in
\emph{typewriter} style.}
```

You can also emphasize text if it is set in italics, in a sans-serif font, or in typewriter style.

제 11 절 환경

```
\begin{environment} text \end{environment}
```

여기서 *environment*는 환경의 이름이다. 환경은 그 호출 순서만 지켜진다면 중첩하여 사용할 수 있다.

```
\begin{aaa}...\begin{bbb}...\end{bbb}...\end{aaa}
```

아래에서 모든 주요 환경들을 설명한다.

11.1 Itemize, Enumerate, Description 환경

Itemize 환경은 단순 나열에, enumerate 환경은 번호 붙은 나열에, description 환경은 사전과 같은 형태의 설명을 위한 나열에 적합하다.

```
\flushleft
\begin{enumerate}
\item You can mix the list
environments to your taste:
\begin{itemize}
\item But it might start to
look silly.
\item[-] With a dash.
\end{itemize}
\item Therefore remember:
\begin{description}
\item[Stupid] things will not
become smart because they are
in a list.
\item[Smart] things, though, can be
presented beautifully in a list.
\end{description}
\end{enumerate}
```

1. You can mix the list environments to your taste:
 - But it might start to look silly.
 - With a dash.
2. Therefore remember:

Stupid things will not become smart because they are in a list.

Smart things, though, can be presented beautifully in a list.

11.2 Flushleft, Flushright, Center 환경

Flushleft와 flushright 환경은 단락을 왼쪽으로 정렬하거나 오른쪽으로 정렬한다. Center 환경은 글을 가운데 정렬한다. \\를 이용해서 줄바꿈을 지정하지 않으면 L^AT_EX이 자동으로 줄바꿈을 결정한다.

```
\begin{flushleft}
This text is\\ left-aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushleft}
```

This text is
left-aligned. L^AT_EX is not trying to make
each line the same length.

```
\begin{flushright}
This text is right-\\aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushright}
```

This text is right-
aligned. L^AT_EX is not trying to make each
line the same length.

```
\begin{center}
At the centre\\of the earth
\end{center}
```

At the centre
of the earth

11.3 Quote, Quotation, Verse 환경

Quote 환경은 인용문, 중요한 구절이나 예문을 표시할 때 이용한다.

```
A typographical rule of thumb
for the line length is:
\begin{quote}
On average, no line should
be longer than 66 characters.
\end{quote}
This is why \LaTeX{} pages have
such large borders by default and
also why multicolumn print is
used in newspapers.
```

A typographical rule of thumb for the line
length is:

On average, no line should be
longer than 66 characters.

This is why L^AT_EX pages have such large bor-
ders by default and also why multicolumn
print is used in newspapers.

이와 유사한 환경으로 quotation과 verse 환경이 있다. quotation 환경은 단락 들여쓰기를 하기 때문에 여러 단락에 걸친 긴 인용문을 처리할 때에 쓴다. verse 환경은 줄바꿈이 중요한 시를 표시할 때에 유용하다. 각 행은 \\로, 각 연은 빈 줄로 구분한다.

```
I know only one English poem by
heart. It is about Humpty Dumpty.
\begin{flushleft}
\begin{verse}
Humpty Dumpty sat on a wall:\
Humpty Dumpty had a great fall.\
All the King's horses and all
the King's men\
Couldn't put Humpty together
again.
\end{verse}
\end{flushleft}
```

I know only one English poem by heart. It is about Humpty Dumpty.

Humpty Dumpty sat on a wall:
Humpty Dumpty had a great
fall.
All the King's horses and all the
King's men
Couldn't put Humpty together
again.

11.4 Abstract 환경

학술 논문에서는 독자들에게 본문의 내용을 미리 알 수 있는 요약글의 처음에 두는 것이 관행이다. 이를 위해 \LaTeX 은 `abstract` 환경을 제공한다. `abstract` 환경은 보통 `article` 클래스로 조판하는 문서에서 사용된다.

```
\begin{abstract}
The abstract abstract.
\end{abstract}
```

The abstract abstract.

11.5 Verbatim 환경

`\begin{verbatim}`와 `\end{verbatim}` 사이에 포함된 텍스트는 마치 타자를 친 것과 같이 \LaTeX 명령이 전혀 수행되지 않고 모든 줄바꿈과 공백이 입력된 그대로 출력된다.

개별 단락 안에서는 다음의 명령으로 비슷한 효과를 낼 수 있다.

```
\verb+text+
```

+는 구분자의 한 예일 뿐으로, *와 공백 문자를 제외한 다른 문자들도 사용할 수 있다. 이 책자에 실린 많은 \LaTeX 예들이 이 명령을 이용하여 조판되었다.

The `\verb|\ldots|` command `\ldots`

```
\begin{verbatim}
10 PRINT "HELLO WORLD ";
20 GOTO 10
\end{verbatim}
```

The `\ldots` command ...

```
10 PRINT "HELLO WORLD ";
20 GOTO 10
```

```
\begin{verbatim*}
the starred version of
the      verbatim
environment emphasizes
the spaces in the text
\end{verbatim*}
```

```
the_starred_version_of
the_verbatim
environment_emphasizes
the_spaces_in_the_text
```

`\verb` 명령도 위의 `\verbatim*` 명령처럼 `*`과 함께 쓰일 수 있다.

```
\verb*|like this :-)|
```

```
like_this_:-)|
```

`verbatim` 환경과 `\verb` 명령은 다른 명령의 인자 안에서 사용할 수 없다.

11.6 Tabular 환경

`tabular` 환경은 보기 좋은 표를 만드는 데 쓰인다. 원한다면 가로줄이나 세로줄도 그릴 수 있다. 표의 열의 폭은 \LaTeX 이 자동으로 조정한다.

`tabular` 환경은 다음과 같이 시작한다.

```
\begin{tabular}[pos]{table spec}
```

위에서 `table spec` 인자는 표의 모양을 지정한다. `[l]`은 각열을 왼쪽 정렬하고, `[r]`은 오른쪽 정렬하며, `[c]`는 가운데 정렬한다. 줄바꿈을 하면서 양쪽 정렬을 하려면 `p{width}`를 이용한다. `[l]`은 세로줄을 그리게 한다.

열 안의 텍스트가 지정된 페이지 폭에 비해 너무 넓을 때, \LaTeX 은 자동으로 줄을 넘겨 주지 않는다. `p{width}`를 이용하면 보통 단락에서처럼 자동으로 줄바꿈이 되는 특별한 열을 정의할 수 있다.

`pos` 인자는 주변 텍스트의 베이스라인을 기준으로 상대적인 수직 위치를 지정한다. `[t]`, `[b]`, `[c]`를 이용해서 표의 위치를 위(top), 아래(bottom), 가운데(center)로 지정할 수 있다.

`tabular` 환경 안에서 `&`는 다음 열로 넘김, `\\`는 새로운 행의 시작, `\hline`은 가로줄 삽입에 쓰인다. `\cline{j-i}`를 이용하면 가로줄을 부분적으로 그릴 수 있다. 여기서 `i`와 `j`는 줄이 시작하는 열과 끝나는 열의 번호이다.

```
\begin{tabular}{|r|l|}
\hline
7C0 & hexadecimal \\
3700 & octal \\ \cline{2-2}
11111000000 & binary \\
\hline \hline
1984 & decimal \\
\hline
\end{tabular}
```

7C0	hexadecimal
3700	octal
11111000000	binary
1984	decimal


```
\begin{tabular}{|p{4.7cm}|}
\hline
Welcome to Boxy's paragraph.
We sincerely hope you'll
all enjoy the show.\\
\hline
\end{tabular}
```

Welcome to Boxy's paragraph. We sincerely hope you'll all enjoy the show.
--

열 구분자는 `@{...}` 형식으로 지정할 수 있다. 이 명령은 열 사이의 공백을 없애고 이를 중괄호 사이에 인자로 주어진 문자열로 바꾼다. 이 문자열에는 특별한 제한이 없다. 이 명령은 흔히 아래에서 설명하는 소수점 정렬에 이용한다. 또 다른 응용 가능성은 `@{}`를 이용해서 표 안의 안쪽 여백을 없애는 것이다.

```
\begin{tabular}{@{} l @{}}
\hline
no leading space\\
\hline
\end{tabular}
```

<u>no leading space</u>

```
\begin{tabular}{l}
\hline
leading space left and right\\
\hline
\end{tabular}
```

<u>leading space left and right</u>

LaTeX에는 숫자가 포함된 열을 소수점 정렬하는 내부 명령이 없으므로²⁷ 두 개의 열을 이용하는 “짚수”를 쓸 수 있다. 즉, 한 열에 오른쪽 정렬된 정수 부분을 두고 또 다른 한 열에는 왼쪽 정렬된 소수부를 두는 것이다. 그리고 `\begin{tabular}` 줄에 `@{.}` 명령을 두어서 열 사이의 공백을 “.”로 바꾸어 하나의 소수점 정렬된 열처럼 보이게 하는 것이다. 열 구분자(&)로 정수부와 소수부를 분리하는 것을 잊으면 안 된다. 이렇게 만든 숫자 “열”의 제목은 `\multicolumn` 명령으로 붙일 수 있다.

```
\begin{tabular}{c r @{.} l}
Pi expression & & \\
\multicolumn{2}{c}{Value} \\
\hline
$\pi$ & 3&1416 & \\
$\pi^{\pi}$ & 36&46 & \\
$(\pi^{\pi})^{\pi}$ & 80662&7 & \\
\end{tabular}
```

Pi expression	Value
π	3.1416
π^{π}	36.46
$(\pi^{\pi})^{\pi}$	80662.7

²⁷사용자 시스템에 ‘tools’ 묶음이 설치되어 있으면 dcolumn 패키지를 살펴 보라.

```
\begin{tabular}{|c|c|}
\hline
\multicolumn{2}{|c|}{Ene} \\
\hline
Mene & Muh! \\
\hline
\end{tabular}
```

Ene	
Mene	Muh!

`tabular` 환경으로 조판된 내용은 한 쪽 안에 모여 나뉘어지지 않는다. 긴 표를 조판하려면 `longtable` 환경을 쓸 수 있다.

제 12 절 떠다니는 개체

오늘날 출판물들은 대부분 많은 그림과 표를 포함하고 있다. 이런 요소들은 여러 쪽에 걸쳐 나뉠 수 없기 때문에 특별히 다루어야 한다. 한 가지 방법은 그림이나 표가 현재 쪽에 들어가기에 너무 클 경우마다 새 쪽을 시작하는 것이다. 이렇게 하면 보기 흉하게 빈 공간이 너무 많은 쪽들이 생겨난다.

이 문제를 해결하는 방법은 현재 쪽에 들어가지 않는 그림이나 표를 “떠다니게” 해서 다음 쪽으로 넘기고 현재 쪽은 본문으로 채우는 것이다. \LaTeX 은 표와 그림이라는 두 가지 떠다니는 개체 환경을 제공한다. 이들 환경을 최대한 유용하게 이용하려면 \LaTeX 이 떠다니는 개체를 내부적으로 어떻게 처리하는지 대강이라도 알고 있어야 한다. 그렇지 않으면, \LaTeX 이 떠다니는 개체를 사용자가 원하는 위치에 절대로 배치하지 않기 때문에 큰 혼란에 빠질 수 있다.

먼저 \LaTeX 이 제공하는 떠다니는 개체를 위한 명령들을 살펴 보자.

`Figure`나 `table` 환경으로 둘러 싸인 내용은 무엇이든지 떠다니는 개체로 취급된다. 두 환경 모두 위치 지정자(*placement specifier*)라고 하는 선택 인자를 지원한다.

```
\begin{figure}[placement specifier] or \begin{table}[placement specifier]
```

이 인자는 \LaTeX 에게 떠다니는 개체가 옮겨갈 수 있는 위치를 알려 준다. 위치 지정자는 떠다니는 개체 배치 허용 범위를 나타내는 문자열로 구성한다. 표 2.7을 보라.

예를 들면, 표는 다음과 같이 시작할 수 있다.

```
\begin{table}[!hbp]
```

위치 지정자 `[!hbp]`는 \LaTeX 에게 이 표를 바로 이곳(h), 쪽의 아래(b), 아니면 특별한 떠다니는 개체 쪽(p)에 배치하도록 한다. 또한 이렇게 배치할 때에 모양이 보기에 좋지 않더라도 상관하지 않는 것이다.(!) 위치 지정자를 명시하지 않으면 기본값으로 `[tbp]`를 사용한다.

L^AT_EX은 모든 떠다니는 개체를 사용자가 지정한 위치 지정자에 따라 배치할 것이다. 만약 현재 쪽에 개체를 배치할 수 없으면, 그림 대기열이나 표 대기열에 들어가게 된다.²⁸ L^AT_EX은 먼저 대기열에 있는 개체들로 특별한 ‘떠다니는 개체’ 쪽을 채울 수 있는지 알아 본다. 이것이 불가능하면 각각의 대기열의 맨처음에 있는 개체들이 현재 위치에 나온 것으로 간주하여 이들을 각각의 위치 지정자(더 이상 적용이 불가능한 ‘h’는 무시됨)에 따라 배치하려 한다. 본문에 나타나는 새 개체들은 모두 해당 대기열로 보내진다. L^AT_EX은 각 떠다니는 개체의 출현 순서를 엄격히 유지한다. 이런 이유로 해서 배치할 수 없는 그림 하나가 그 뒤에 나온 모든 그림들을 문서 맨끝으로 밀어버리는 현상이 발생하는 것이다. 그러므로,

L^AT_EX이 사용자가 원하는 대로 떠다니는 개체들을 배치하지 못할 때에는, 단지 한 개의 개체가 두 개의 떠다니는 개체 대기열 가운데 하나를 꼭 막고 있는 경우가 많다.

L^AT_EX에 위치 지정자를 한 개만 줄 수도 있지만, 이는 문제를 일으킨다. 떠다니는 개체 하나가 지정된 위치에 배치될 수 없으면 뒤에 나오는 개체들을 막으면서 꼼짝달싹 못하게 된다. 특히 [h] 선택 사항은 절대로 사용해서는 안 된다. 이는 문제가 너무 커서 L^AT_EX 최신판은 [h]를 자동으로 [ht]로 바꾼다.

이제 어려운 사항들에 관한 설명은 마쳤고, 몇가지 남아 있는 `table`과 `figure` 환경에 관한 사항들을 살펴 보자.

²⁸이들은 먼저 들어간 것이 먼저 나오는 (FIFO — ‘first in first out’) 큐(queue)이다.

표 2.7: 떠다니는 개체 배치 허용 범위

지정자	떠다니는 개체 배치 허용 범위...
<code>h</code>	<i>here</i> 를 의미. 떠다니는 개체 명령이 쓰인 바로 그 자리. 작은 개체들에 주로 쓰임.
<code>t</code>	<i>top</i> 을 의미. 쪽의 맨 위.
<code>b</code>	<i>bottom</i> 을 의미. 쪽의 맨 아래.
<code>p</code>	<i>page</i> 를 의미. 떠다니는 개체만 모이는 특별한 쪽에 배치.
<code>!</code>	떠다니는 개체의 위치를 제어하는 내부 변수들 ^a 을 고려하지 않도록 함. 해당 개체가 배치되지 않을 수도 있음.

`pt`, `em`는 T_EX에서 사용하는 단위이다. 이들에 대해서는 118 쪽의 표 6.5를 보라.

^a한 쪽에 올 수 있는 개체의 최대값 등

아래의 명령을 이용해서 떠다니는 개체의 캡션을 정의할 수 있다.

```
\caption{caption text}
```

일련 번호와 “그림” 혹은 “표” 문자열은 L^AT_EX이 붙여준다.

```
\listoffigures and \listoftables
```

위의 두 명령은 `\tableofcontents` 명령과 마찬가지로 작동하는데, 각각 그림 차례와 표 차례를 출력한다. 차례에는 캡션이 잘리지 않은 채로 전체가 나타나므로, 긴 캡션을 사용하려면 차례에 표시될 짧은 캡션을 따로 만들어야 한다. 짧은 캡션을 `\caption` 명령 다음에 각괄호 안에 써주면 된다.

```
\caption[Short]{LLLLLooooooooonnnnnnggggg}
```

`\label`과 `\ref`로 문서 내의 떠다니는 개체를 참조할 수 있다.

다음의 예는 정사각형을 하나 그린 다음, 그것을 문서에 삽입하는 것을 보이고 있다. 이러한 방법으로 완성된 문서에 포함할 그림을 위한 공간을 미리 확보할 수 있다.

```
Figure~\ref{white} is an example of Pop-Art.
\begin{figure}[!hbp]
\makebox[\textwidth]{\framebox[5cm]{\rule{0pt}{5cm}}}
\caption{Five by Five in Centimetres.\label{white}}
\end{figure}
```

위의 예에서, L^AT_EX은 그림을 바로 그 자리(h)에 배치하려고 최선을 다할 것이다.²⁹ 만일 그것이 불가능하다면, 그림을 현재 쪽의 바닥(b)에 놓으려 할 것이다. 현재 쪽에 그림을 배치하는 것이 전혀 불가능하면, L^AT_EX은 이 그림과 대기열에서 대기 중인 표가 배치된 떠다니는 개체를 위한 쪽을 만들 수 있는지의 여부를 결정한다. 특별한 떠다니는 개체를 위한 면을 채울 만큼 표나 그림이 충분하지 않다면 L^AT_EX은 새 쪽을 만들면서 다시 한 번 이 그림을 텍스트의 바로 그 위치에서 처음 나타난 것으로 간주한다.

어떤 특별한 상황에서는 다음의 명령을 사용해야 할 수도 있다.

```
\clearpage or even the \cleardoublepage
```

이들 명령은 L^AT_EX로 하여금 대기 중인 모든 떠다니는 개체를 즉시 출력하고 새 쪽을 시작하도록 한다. `\cleardoublepage`는 빈 쪽을 하나 만들어서라도 새 쪽으로 간다.

L^AT_EX 2_ε 문서에 POSTSCRIPT 그림을 삽입하는 방법은 이 책자의 뒷부분에서 배울 수 있다.

²⁹ 그림 대기열이 비어 있다고 가정할 때

제 13 절 풀리는 명령 보호하기

`\caption`이나 `\section`와 같은 명령의 인자로 사용한 텍스트는 한 문서 안에서 두 번 이상 식자되는 경우가 있다. 예를 들어, 같은 텍스트가 문서의 본문에도 나타나고 차례에도 나타날 수가 있는 것이다. 어떤 명령은 `\section` 등의 장절 명령의 인자 안에서 사용되면 정상적으로 작동하지 않는다. 이러한 명령을 풀리는 명령이라고 하는데, `\footnote`, `\phantom`이 그 예이다. 이러한 풀리는 명령은 풀림을 방지하기 위한 보호가 필요하다. 풀리는 명령을 보호하려면 해당 명령 앞에 `\protect`를 써주면 된다.

`\protect`는 바로 뒤따르는 명령에만 효력이 있고, 그 인자에는 영향을 미치지 않는다. 대부분의 경우 `\protect` 명령은 남발하여도 별다른 문제가 생기지 않는다.

```
\section{I am considerate
        \protect\footnote{and protect my footnotes}}
```


제 3 장

수식 조판하기

이제 준비가 됐다! 이 장에서 우리는 T_EX의 강력한 기능의 하나인 수식 조판을 공략할 것이다. 하지만 이 장은 그저 수박 겉핥기라는 점을 일러 둔다. 여기서 설명하는 것만으로도 많은 이들에게 충분하겠지만, 원하는 수식 조판의 답을 여기서 찾지 못했다고 좌절하진 말기 바란다. [AMS-L^AT_EX¹](#)에 해답이 있을 확률이 높다.

제 1 절 개괄

L^AT_EX은 수학적 조판을 위한 특별한 모드를 갖고 있다. 수학적식은 단락 안에서 행중(inline) 수식으로 조판될 수도 있고, 별도의 단락으로 조판될 수도 있다. 단락 안의 행중 수식 텍스트는 `\`(와 `\`) 사이, `$`와 `$` 사이, 또는 `\begin{math}`와 `\end{math}` 사이에 들어간다.

Add `a` squared and `b` squared
to get `c` squared. Or, using
a more mathematical approach:
`$c^2=a^2+b^2$`

Add *a* squared and *b* squared to get *c*
squared. Or, using a more mathematical ap-
proach: $c^2 = a^2 + b^2$

`\TeX{}` is pronounced as
`\(\tau\epsilon\chi\)`.
100 `m^3` of water
This comes from my
`\begin{math}\heartsuit\end{math}`

T_EX is pronounced as τϵχ.
100 m³ of water
This comes from my ♥

¹미국수학회에서는 L^AT_EX의 강력한 확장판을 만들었다. 이 장에서 많은 예제들은 이 확장판을 쓴다. 최근에 나온 모든 T_EX 배포판에 이 확장판이 들어 있다. 만약 빠져 있다면 [CTAN: /tex-archive/macros/latex/required/amslatex](#)에 가 보자.

별도 단락으로 큰 수학 기호를 사용하는 방정식등의 수학식을 식자하려면 텍스트 모드에서 단락을 나누듯이 새로운 단락을 시작하지 말고 수식 보여주기(display)하는 것이 좋다. 이렇게 하기 위해서, 이들을 `\[`와 `\]`안에 넣거나, `\begin{displaymath}`와 `\end{displaymath}` 사이에 넣는다.²

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
\begin{displaymath}
c^2=a^2+b^2
\end{displaymath}
or you can type less with:
\[a+b=c\]
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach:

$$c^2 = a^2 + b^2$$

or you can type less with:

$$a + b = c$$

L^AT_EX에서 수식 번호를 자동으로 매겨주기를 원한다면, `equation` 환경을 쓸 수 있다. 그러면 수식 번호에 `\label`을 붙이고 `\ref` 또는 `amsmath` 패키지의 `\eqref` 명령을 써서 본문 안 다른 곳에서 이 레이블을 참조할 수 있다.

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
From (\ref{eq:eps}), we gather
\ldots{}From \eqref{eq:eps} we
do the same.
```

$$\epsilon > 0 \quad (3.1)$$

From (3.1), we gather ... From (3.1) we do the same.

행중(inline) 수식과 수식 보여주기(display) 수식 사이에는 조판 스타일에 차이가 있다는 점을 유의하자.

```
 $\lim_{n \to \infty}$
 $\sum_{k=1}^n \frac{1}{k^2}$
 = $\frac{\pi^2}{6}$
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

수학 모드와 텍스트 모드는 다르다. 예를 들면, 다음은 수학 모드의 특징 가운데 일부이다.

²plainT_EX에서는 `$$`와 `$$` 사이에 수식을 넣어서 수식 보여주기를 구현한다. 그러나 이 방법은 시작과 끝을 명확하게 파악하기 어려운 경우가 있으므로 L^AT_EX 방식을 사용하는 것이 좋을 것이다. —[역자]

1. 빈 칸과 줄바꿈은 아무런 의미도 없다. 수식의 공백은 수학식 표현의 논리에 따라 정해지며 이것을 제어하려면 `\,`, `\quad` 또는 `\qquad`와 같은 특별한 명령으로 지정해야 한다.
2. 빈 줄은 허용되지 않는다. 하나의 수식을 여러 문단으로 적을 수 없다.
3. 각 글자는 변수명으로 간주되어 변수로 조판될 것이다. 일반 텍스트(정상적인 곧게 선 글꼴과 일반적인 간격)를 수식 안에서 조판하고자 한다면 텍스트를 `\text{...}` 명령 안에 넣어야 한다 (59페이지의 제 7 절도 참조할 것).³

```
\begin{equation}
\forall x \in \mathbf{R}:
\quad x^2 \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \quad x^2 \geq 0 \quad (3.2)$$

```
\begin{equation}
x^2 \geq 0 \quad \text{for all } x \in \mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbf{R} \quad (3.3)$$

기호문자 선택에 무척 까다로운 수학자들이 있다. 위의 경우 실수 집합을 나타내는 데 굵은 글씨 **R**을 썼는데 이것을 ‘블랙보드 볼드’(blackboard bold)를 쓰는 관행이 있다. 이 기호는 `amsmath` 또는 `amssymb` 패키지에 있는 `\mathbb{R}` 명령을 써서 얻을 수 있다.

그러므로 예제 마지막 줄은 다음과 같이 한다.

```
\begin{displaymath}
x^2 \geq 0 \quad \text{for all } x \in \mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbb{R}$$

제 2 절 수학 모드에서 묶기

대부분 수학 모드 명령들은 바로 다음 한 글자에 대해서만 유효하기 때문에, 만약 명령이 여러 글자에 영향을 미치기를 원한다면 이들을 중괄호를 써서 묶어주어야 한다: `{...}`.

```
\begin{equation}
a^{x+y} \neq a^{x+y}
\end{equation}
```

$$a^{x+y} \neq a^{x+y} \quad (3.4)$$

³`amsmath` 패키지의 `\text` 명령은 수식 모드 안에서 텍스트 모드로 빠져나가도록 해준다. 이 밖에 `\mbox`를 이용하는 방법도 있다. —[역주]

제 3 절 수식 구성하기

이 절에서는 수식 조판에 쓰이는 가장 중요한 명령들을 설명한다. 수학 기호 조판을 위한 명령의 자세한 목록을 보려면 63 쪽 제 10 절을 살펴보자.

그리스 소문자는 `\alpha`, `\beta`, `\gamma`, ... 와 같이 입력하고 대문자들은 `\Gamma`, `\Delta`, ... 와 같이 입력한다.⁴

`\lambda, \xi, \pi, \mu, \Phi, \Omega`

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

위첨자와 아래첨자는 `^`와 `_` 문자를 써서 지정한다.

`a_{1} \quad x^{2} \quad`
`$e^{-\alpha t}$ \quad`
`a^{3}_{ij} \quad`
`$e^{x^2} \neq {e^x}^2$`

$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$
 $e^{x^2} \neq e^x^2$

제곱근은 `\sqrt`로 입력한다. n차 제곱근은 `\sqrt[n]` 형식으로 쓰면 된다. 제곱근 부호의 크기는 L^AT_EX이 자동으로 판단한다. 근호만이 필요할 경우라면, `\surd`를 쓰도록 하자.

`\sqrt{x} \quad`
`$\sqrt{x^2+\sqrt{y}}$ \quad`
`\quad $\sqrt[3]{2}$ \quad [3pt]`
`$\surd[x^2 + y^2]$`

$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$
 $\sqrt{[x^2 + y^2]}$

`\overline`과 `\underline` 명령은 수식의 바로 위나 아래에 수평선을 긋는다.

`$\overline{m+n}$`

$\overline{m+n}$

`\overbrace`와 `\underbrace` 명령은 수식의 위나 아래에 긴 수평 중괄호를 만든다.

`$\underbrace{a+b+\cdots+z}_{26}$`

$\underbrace{a+b+\cdots+z}_{26}$

작은 화살표나 물결표 기호 같은 수식 액센트를 변수에 추가하려면 63 페이지에 있는 표 3.1에 있는 명령을 쓴다. `\widehat` 그리고 `\widetilde` 명령으로 여러 글자에 걸친 넓은 모자 기호나 물결표를 만들 수 있다. '는 프라임 부호를 붙여준다.

⁴L^AT_EX 2_ε에는 대문자 알파가 정의되어 있지 않는데 로마자 A와 모양이 같기 때문이다. 새로운 수학 부호계(new math coding)를 사용하도록 하면 바뀔 것이다.

```
\begin{displaymath}
y=x^2\quad y'=2x\quad y''=2
\end{displaymath}
```

$$y = x^2 \quad y' = 2x \quad y'' = 2$$

벡터는 종종 변수의 위에 작은 화살표 부호를 더하여 나타낸다. 이는 `\vec` 명령으로도 할 수 있다. 두 명령 `\overrightarrow`와 `\overleftarrow`는 A 에서 B 로 가는 벡터를 표시할 때 쓴다.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec{a} \quad \overrightarrow{AB}$$

보통은 곱셈 연산을 가리키기 위한 도트 기호를 명시적으로 조판하지 않는다. 하지만 때때로 공식을 그룹으로 묶어서 독자가 알아보기 쉽게 만들기 위해서 이 기호를 쓰기도 한다. 이런 경우에는 `\cdot`를 써야 한다.

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2 \tau_1 \cdot \tau_2
\end{displaymath}
```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

로그(log)류 함수명은 대개 변수에 적용되는 이탤릭체가 아닌 곧게 선 글꼴로 조판된다. \LaTeX 은 다음과 같은 명령을 제공하여 중요한 함수명 식자에 사용하도록 하고 있다.

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

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

모듈로 함수(modulo function)를 위해서는 두 가지 명령이 있다. 이항 연산에 해당하는 `\bmod` 명령과 “ $x \equiv a \pmod{b}$ ”와 같은 표현을 위한 `\pmod`가 그것이다.

```
$a\bmod b$
$x\equiv a \pmod{b}$
```

$$a \bmod b \\ x \equiv a \pmod{b}$$

분모와 분자를 상하로 배열하는 **분수**는 `\frac{...}{...}` 명령으로 조판된다. 종종 $1/2$ 과 같이 사선을 그어 표시하는 형식이 더 나올 수도 있는데, 적은 양의 ‘분수류’에 대해서는 사선 형식이 더 나아보일 때가 있기 때문이다.

```

$1\frac{1}{2}$~hours
\begin{displaymath}
\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}
\end{displaymath}

```

 $1\frac{1}{2} \text{ hours}$

$$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

이항 계수 또는 그와 비슷한 구조를 조판하기 위해서는 `amsmath` 패키지의 `\binom` 명령을 쓰면 된다.

```

\begin{displaymath}
\binom{n}{k} \quad C_n^k
\end{displaymath}

```

$$\binom{n}{k} \quad C_n^k$$

이항 관계식에서 기호를 위 아래에 배치해야 할 때가 있다. `\stackrel`은 두번째 인자를 정상적인 위치에 두고 첫번째 인자로 주어진 기호를 그 윗쪽에 상첨자 크기로 놓는다.

```

\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}

```

$$\int f_N(x) \stackrel{!}{=} 1$$

적분 기호는 `\int`로, **합 기호**는 `\sum`으로, 그리고 **곱 기호**는 `\prod`로만 들어진다. 위끝 아래끝 극한은 위첨자와 아래첨자처럼 `^`과 `_` 기호로 만든다.⁵

```

\begin{displaymath}
\sum_{i=1}^n \quad \int_0^{\pi/2} \quad \prod_{\epsilon}
\end{displaymath}

```

$$\sum_{i=1}^n \quad \int_0^{\pi/2} \quad \prod_{\epsilon}$$

복잡한 수식에서 첨자형 지수의 위치 조절을 위해 `amsmath`는 두 가지 추가 도구를 제공한다. `\substack` 명령과 `subarray` 환경이다.

```

\begin{displaymath}
\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i,j) =
\sum_{\begin{subarray}{l} i \in I \\ 1 < j < m \end{subarray}} Q(i,j)
\end{displaymath}

```

$$\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i,j) = \sum_{\substack{i \in I \\ 1 < j < m}} Q(i,j)$$

\TeX 은 **괄호**와 짝맞춤 문자(예를 들어, $[\langle \parallel \uparrow$)를 위한 대부분의 기호를 제공한다. 둥근 괄호(소괄호)와 각진 괄호(대괄호)는 키보드의 해당 키로부터

⁵여러 줄에 걸친 위첨자와 아래첨자를 쓰려면 $\mathcal{AMS-}\text{\LaTeX}$ 패키지가 필요하다.

그리고 중괄호는 \{로 입력할 수 있으나 그밖의 다른 짝맞춤 기호들은 특별한 명령으로 얻는다 (예를 들면 `\updownarrow` ↓). 사용할 수 있는 모든 짝맞춤 기호의 목록을 65 페이지에 있는 표 3.8에서 찾아볼 수 있다.

```
\begin{displaymath}
\{a,b,c\}\neq\{a,b,c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

여는 짝맞춤 기호 앞에 `\left`, 닫는 짝맞춤 기호 앞에 `\right` 명령을 두면 \TeX 이 짝맞춤 기호의 크기를 적당하게 자동으로 맞추어준다. `\left`를 썼다면 반드시 `\right`로 닫아줘야 한다는 점, 그리고 둘 모두 같은 줄에서 조판이 될 때만 크기가 적절하게 정해진다는 점을 주의하자. 큰 왼쪽 괄호를 얻고 싶으나 오른쪽에는 닫는 괄호를 적지 않아야 하는 경우와 같이 오른쪽 짝맞춤 기호 위치를 비우려면 실제로는 아무것도 식자하지 않는 (논리적인) 오른쪽 짝맞춤 `\right.`를 사용하자.

```
\begin{displaymath}
1 + \left( \frac{1}{1-x^2} \right)^3
\end{displaymath}
```

$$1 + \left(\frac{1}{1 - x^2} \right)^3$$

수학 짝맞춤 기호의 적당한 크기를 수동으로 직접 지정해야 할 때가 있다. 이럴 때는 `\big`, `\Big`, `\bigg` 그리고 `\Bigg` 명령을 접두사로서 짝맞춤 기호 명령 앞에 붙여서 지정할 수 있는 경우가 많다.⁶

$$\frac{(x+1)(x-1)}{x^2}$$

$$\left((x+1)(x-1) \right)^2$$

점 세 개를 수식 안에 넣기 위한 명령이 몇 가지 있다. `\ldots`는 점들을 베이스라인에 식자하며 `\cdots`는 중간 높이에 식자한다. 아울러, 점 세 개를 세로로 놓는 `\vdots`와 점 세 개를 대각선으로 놓는 `\ddots`가 있다. 56 페이지 제5절에서 다른 예제를 찾을 수 있다.

```
\begin{displaymath}
x_{\{1\}}, \ldots, x_{\{n\}} \quad \qquad \qquad
x_{\{1\}} + \cdots + x_{\{n\}}
\end{displaymath}
```

$$x_1, \dots, x_n \qquad x_1 + \dots + x_n$$

⁶이 명령들은 글꼴 크기 변경 명령이 쓰인 경우나 11pt 또는 12pt 선택 사항들이 지정된 경우에는 원하는 대로 작동하지 않는다. 이런 문제점을 바로 잡기 위해서는 exscale 또는 amsmath 패키지를 쓰도록 하자.

제 4 절 수식의 간격

만약 \TeX 이 선택한 수식 내 간격이 만족스럽지 않다면, 특별한 간격 명령을 집어넣어서 조절할 수 있다. 작은 간격을 위한 명령 몇 가지는 다음과 같다. $\backslash, (u)$ 는 $\frac{3}{18}$ quad, $\backslash:(u)$ 는 $\frac{4}{18}$ quad, $\backslash;(u)$ 는 $\frac{5}{18}$ quad 간격을 벌려준다. 강제 공백 문자 $\backslash_$ 는 반각 간격을 벌리며 $\backslashquad(_)$ 와 $\backslashqqquad(_)$ 는 전각 및 2배각의 큰 공백을 만든다.

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\!\!\int\int_D g(x,y)
\quad \backslash, \quad \backslashud x \backslash, \quad \backslashud y
\end{displaymath}
instead of
\begin{displaymath}
\int\!\!\!\!\!\int\int_D g(x,y)\backslashud x \backslashud y
\end{displaymath}
```

$$\iint_D g(x,y) \, dx \, dy$$

instead of

$$\int \int_D g(x,y) \, dx \, dy$$

미분의 ‘d’는 보통 곧은 글꼴로 조판된다는 것에 주의하자.⁷

$\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 은 중적분 기호 사이의 간격을 미세 조정하기 위하여 \backslashiint , \backslashiiint , \backslashiiiint , \backslashidotsint 명령을 제공한다. amsmath 패키지를 로드하면 위 예제를 다음과 같이 조판할 수 있다.

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_D \backslash, \quad \backslashud x \backslash, \quad \backslashud y
\end{displaymath}
```

$$\iint_D dx \, dy$$

더 자세한 내용은 전자 문서 `testmath.tex`($\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 과 함께 배포된다) 또는 *The $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ Companion* [3]의 8장을 보라.

제 5 절 세로 줄맞춤이 필요한 요소들

행렬을 조판하려면, `array` 환경을 쓴다.⁸ 이것은 `tabular` 환경과 비슷하다. `\` 명령으로 줄 나눔을 표시한다.

⁷우리나라의 수학 문헌에서는 일반적으로 미분의 ‘d’도 이탤릭을 그대로 사용하는 예가 많다. —[역주]

⁸ amsmath 에서는 행렬 조판을 위한 `matrix` 류의 환경을 다양하게 제공한다. —[역주]

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \ldots \\
x_{21} & x_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

array 환경은 오른쪽 짝맞춤 문자(\right) 위치에 식자되지 않는 “.”를 써서 한 개의 큰 묶음표를 가지는 수식을 조판하는 데에도 쓸 수 있다.

```
\begin{displaymath}
y = \left\{ \begin{array}{l}
a \quad \text{if } d > c \\
b+x \quad \text{in the morning} \\
l \quad \text{all day long}
\end{array} \right.
\end{displaymath}
```

$$y = \begin{cases} a & \text{if } d > c \\ b+x & \text{in the morning} \\ l & \text{all day long} \end{cases}$$

tabular 환경처럼 array 환경 안에서도 줄을 그을 수 있다. 예를 들어 행렬의 원소를 갈라 놓으려면 다음과 같이 한다.

```
\begin{displaymath}
\left( \begin{array}{c|c}
1 & 2 \\ \hline
3 & 4
\end{array} \right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array} \right)$$

여러 줄에 걸친 식이나 수식군을 위해서는 equation 대신 eqnarray 환경을, equation* 대신 eqnarray*를 쓸 수 있다. eqnarray 안에서 각 줄마다 하나씩 수식번호가 붙는다. eqnarray*는 번호가 붙지 않는다.

eqnarray와 eqnarray* 환경은 {rcl} 형식의 세 컬럼 표(tabular)와 같이 동작하며, 가운데 칸은 등호, 부등호, 그밖에 적당한 다른 기호를 두는 데 쓴다. \\ 명령은 줄 나눔을 표시한다.

```
\begin{eqnarray}
f(x) & = & \cos x & \\
f'(x) & = & -\sin x & \\
\int_0^x f(y)dy & & & \\
& = & \sin x & \\
\end{eqnarray}
```

$$\begin{aligned} f(x) &= \cos x & (3.5) \\ f'(x) &= -\sin x & (3.6) \\ \int_0^x f(y)dy &= \sin x & (3.7) \end{aligned}$$

등호 양쪽 편의 간격이 좀 넓어 보인다는 것을 알아 두자. 다음 예제에서와 같이 \setlength\arraycolsep{2pt}로 설정하면 이를 고칠 수 있다.

긴 수식은 알아서 적절하게 나누어지지 않을 것이다. 저자는 어디서 나눌 것인지 들여쓰기를 얼마로 할지 지정해야 한다. 이를 위해서 다음 두 방법이 가장 널리 쓰인다.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x &= & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
&\quad \quad \quad \nonumber \\
&& \frac{x^7}{7!} + \cdots
\end{eqnarray}}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \quad (3.8)$$

```
\begin{eqnarray}
\lefteqn{ \cos x = 1 } \\
&\quad - \frac{x^2}{2!} + \\
&\quad \quad \quad \nonumber \\
&& \frac{x^4}{4!} - \\
&\quad - \frac{x^6}{6!} + \cdots
\end{eqnarray}}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \quad (3.9)$$

`\nonumber` 명령은 L^AT_EX에게 이 수식에 번호를 붙이지 말라고 알려준다.

이런 방법들만으로 세로로 똑바르게 정렬 배치되는 수식을 얻기 어려울 수도 있다. `amsmath` 패키지는 강력한 대안을 제공하는데, `align`, `falign`, `gather`, `multline`, `split` 환경이 그것이다. 일별해보도록 하라.

제 6 절 허깨비들

허깨비는 눈에 보이지 않지만 여전히 많은 사람들의 마음 속에 한 자리를 차지하고 있다. L^AT_EX도 다를 바 없다. 눈에 보이지 않는 공백 처리를 위해 허깨비(`phantom`)를 응용할 수 있다.

`~`와 `_`를 써서 텍스트를 세로로 정렬할 때 L^AT_EX에게 맡겨두면 결과가 반드시 원하지대로 나오지 않는 수가 있다. `\phantom` 명령을 쓰면 최종 출력에서 보이지 않는 글자들만큼 빈 공간을 확보할 수 있다. 다음 예제를 보면 쉽게 이해가 될 것이다.

```
\begin{displaymath}
{}^{12}_{6}\text{C} \phantom{{}^{12}_{6}\text{C}} \text{versus} \phantom{{}^{12}_{6}\text{C}} {}^{12}_{6}\text{C}
\end{displaymath}
```

$${}^{12}_{6}\text{C} \quad \text{versus} \quad {}^{12}_{6}\text{C}$$


```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k}
\quad\quad\quad\text{versus}\quad\quad\quad\Gamma_{ij}^k
\end{displaymath}
```

$$\Gamma_{ij}^{k} \quad \text{versus} \quad \Gamma_{ij}^k$$

첨자 위치의 이동에 주의하여 보라.

제 7 절 수학 글꼴 크기

수학 모드에서 \LaTeX 은 글자 크기를 문맥에 따라 스스로 선택한다. 예를 들어, 위첨자는 작은 글꼴로 식자된다. 수식의 한 부분을 곧게 선 글꼴로 하고 싶다면 `\textrm` 명령을 쓰면 안되는데, `\textrm` 명령은 임시로 텍스트 모드로 빠져 나오기 때문에 글꼴 크기 전환 메커니즘이 작동하지 않기 때문이다. 크기 전환 메커니즘을 계속 유지하려면 대신 `\mathrm`을 쓰도록 하자. 하지만 `\mathrm`이 제대로 동작하는 것은 짧은 범위에 한정된다는 데 주의하라. 공백은 여전히 전환되지 않고 액센트 붙은 문자에는 통하지 않는다.⁹

```
\begin{equation}
2^{\textrm{nd}} \quad \quad \quad 2^{\mathrm{nd}}
\end{equation}
```

$$2^{\text{nd}} \quad 2^{\text{nd}} \quad (3.10)$$

그래도 \LaTeX 에게 올바른 글꼴 크기를 알려줘야 하는 때가 가끔 있다. 수학 모드에서는 다음 네 가지 명령으로 이를 설정한다.

```
\displaystyle (123), \textstyle (123), \scriptstyle (123) and
\scriptscriptstyle (123).
```

스타일을 바꾸면 상/하한값이 표시되는 방식도 달라진다.

```
\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
\frac{\displaystyle
\sum_{i=1}^n(x_i-\overline{x})
(y_i-\overline{y})}
{\displaystyle\biggl[
\sum_{i=1}^n(x_i-\overline{x})^2
\sum_{i=1}^n(y_i-\overline{y})^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\text{corr}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2}}$$

이것은 `\left[\right]`에 의한 것보다 더 큰 대괄호를 만드는 보기이다. 왼쪽 오른쪽 대괄호 위치에 각각 `\biggl`과 `\biggr` 명령이 쓰였다.

⁹ \LaTeX (`\amsmath`) 패키지를 쓰면 `\textrm` 명령이 크기 변경과 함께 작동하도록 한다.

제 8 절 정리, 법칙, ...

수학 문서를 작성할 때, “보조 정리(Lemma)”, “정의(Definition)”, “공리(Axiom)” 류의 구조를 조판하는 방법을 필요로 할 경우가 있다. L^AT_EX은 이에 해당하는 명령을 지원한다.

```
\newtheorem{name}[counter]{text}[section]
```

name 인자는 “정리”를 식별하기 위한 짧은 키워드다. *text* 인자로는 최종 문서에서 인쇄될 “정리”의 실제 이름을 정의한다.

대괄호 안에 있는 인자들은 선택 사항이다. 둘 다 “정리”에서 번호를 매기는 방법을 지정하는 데에 쓰인다. *counter* 인자는 이전에 정의된 “정리”의 명칭(*name*)으로 지정하도록 하라. 그러면 새로 정의된 정리가 이전 “정리”의 일련 번호를 이어받아 매겨질 것이다. *section* 인자는 “정리” 번호가 어떤 장 절 단위의 번호에 연동되도록 할 것인지 지시하는 것이다. 이것을 *section* 으로 하면 새로운 절이 시작될 때 번호를 새로 시작하면서 정리의 번호를 “절 번호. 정리번호” 형식으로 식자해준다.

전처리부에서 `\newtheorem` 명령을 선언한 후 문서에서 다음 명령을 쓸 수 있다.

```
\begin{name}[text]
This is my interesting theorem
\end{name}
```

설명은 이 정도면 될 것이다. 다음 예제들을 통하여 남아 있을 의문을 없애고 복잡해보이는 `\newtheorem` 환경의 용법을 명확히 하도록 하자.

```
% definitions for the document
% preamble
\newtheorem{law}{Law}
\newtheorem{jury}[law]{Jury}
%in the document
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}\end{jury}
\begin{law}No, No, No\end{law}
```

Law 1 *Don't hide in the witness box*

Jury 2 (The Twelve) *It could be you! So beware and see law **1***

Law 3 *No, No, No*

“Jury” 정리는 “Law” 정리와 같은 카운터를 쓰게 설정되었다. 그러므로 “Jury”의 정리 번호는 이전의 “Law” 정리 번호에 이어진다. 정리 환경의 대괄호 안 옵션 인자는 정리의 제목이나 표제를 특정하는 데 쓴다.

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
If there are two or more
ways to do something, and
one of those ways can result
in a catastrophe, then
someone will do it.\end{mur}
```

Murphy 8.1 *If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.*

“머피” 정리는 현재 절 번호와 연동되어 있다. 이 자리에는 chapter나 sub-section과 같은 장절 단위를 사용할 수 있다.

제 9 절 수학 볼드 기호

L^AT_EX에서 두꺼운 기호를 식자하는 것이 쉽지 않다. 아마도 아마추어 조판가들이 이를 남용하려는 경향이 있어서 일부러 그런 것 같다. 글꼴을 바꾸는 명령인 `\mathbf`로 두툼한 글자를 만들 수 있지만, 수학 기호들이 보통 이탤릭인 데 반해 이렇게 하면 로만(곧게 선) 글꼴이 된다. `\boldmath` 명령이 있지만, 이 명령은 수학 모드 밖에서만 쓸 수 있으므로 텍스트 모드로 빠져나가서 사용해야 한다. 그리고 문장부호에도 함께 적용된다.

```
\begin{displaymath}
\mu, M \quad \mathbf{M} \quad \quad \mu, M
\mbox{\boldmath $\mu, M$}
\end{displaymath}
```

$$\mu, M \quad \mathbf{M} \quad \mu, M$$

셈표까지 굵은 기호가 되어 있는데, 이것은 원하는 바가 아닐 수도 있다는 것에 유의하자.

`amsbsy` 패키지(`amsmath` 패키지에 포함되어 있다), 또는 `tools` 패키지 묶음에 들어 있는 `bm` 패키지를 쓰면 `\boldsymbol` 명령으로 훨씬 쉽게 이 문제를 해결하게 해준다.

```
\begin{displaymath}
\mu, M \quad \quad \mu, M
\boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}$$

이밖에 `amsbsy` 패키지의 `\pmb` 명령은 이른바 “가난한 사람을 위한 굵은 수학 기호”(Poorman’s bold)를 이용하여 원하는 대부분의 문자를 볼드로 만들어준다. 이 ‘가짜 볼드’가 마음에 들지 않는다면 볼드 수학 기호를 포함하고 있는 수학 글꼴 묶음을 찾아보는 방법도 있다. 예를 들어 `mathdesign`의 글꼴들 가운데 볼드 수학 기호를 포함하고 있는 것도 있다. 그러나 이 때는 익숙한 CM의 수학 글꼴을 사용할 수 없게 된다. 다음은 ‘가짜 볼드’를 사용한 예이다.

```
\begin{displaymath}
\mu, M \quad \quad
\pmb{\mu}, \pmb{M}
\end{displaymath}
```

$$\mu, M \quad \quad \pmb{\mu}, \pmb{M}$$

제 10 절 수학 기호 일람

다음 표들은 수학 모드에서 사용할 수 있는 기호들을 보여준다.

표 3.12-3.16에 열거된 기호를 사용하려면¹⁰ `amssymb` 패키지를 문서의 전 처리부에 선언하여야 하고 AMS 수학 글꼴이 시스템에 설치되어 있어야 한다. 만약 AMS 패키지와 글꼴이 설치되어 있지 않다면 `macros/latex/amslatex` 을 찾아보라. 기호문자에 대한 완전한 목록은 `info/symbols/comprehensive` 에서 볼 수 있다.

표 3.1: 수학 모드 악센트

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

표 3.2: 그리스 소문자

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

표 3.3: 그리스 대문자

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

¹⁰이 표들은 David Carlisle 씨가 처음 작성하고 그후 특히 Josef Tkadlec 씨의 제안으로 크게 확장된 `symbols.tex`에서 가져온 것이다.

표 3.4: 이항 관계 연산자

다음 기호들은 그 앞에 `\not` 명령을 붙여서 부정형 기호를 만들 수 있다.

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

^a이 기호를 표시하려면 `latexsym` 패키지가 필요하다.

표 3.5: 이항 연산자

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

표 3.6: 큰 연산자

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

표 3.7: 화살표

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

^a이 기호를 표시하려면 `latexsym` 패키지가 필요하다.

표 3.8: 짝맞춤 기호

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

표 3.9: 큰 짝맞춤 기호

$\left($	<code>\lgroup</code>	$\right)$	<code>\rgroup</code>	$\left[$	<code>\lmoustache</code>	$\right]$	<code>\rmoustache</code>
\uparrow	<code>\arrowvert</code>	\Uparrow	<code>\Arrowvert</code>	\uparrow	<code>\bracevert</code>		

표 3.10: 기타 기호

...	<code>\dots</code>	...	<code>\cdots</code>	⋮	<code>\vdots</code>	⋯	<code>\ddots</code>
\hbar	<code>\hbar</code>	\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>	\wp	<code>\wp</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\mho ^a	<code>\mho</code>	∂	<code>\partial</code>
'	<code>'</code>	'	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>	\Box ^a	<code>\Box</code>	\diamond	<code>\Diamond</code> ^a
\bot	<code>\bot</code>	\top	<code>\top</code>	\angle	<code>\angle</code>	\surd	<code>\surd</code>
\diamondsuit	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg or \nmid	<code>\neg</code> or <code>\nmid</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>

^a이 기호를 표시하려면 `latexsym` 패키지가 필요하다.

표 3.11: 수학 기호가 아닌 것

이 기호들은 텍스트 모드에서도 사용가능하다.

\dagger	<code>\dag</code>	\S	<code>\S</code>	\copyright	<code>\copyright</code>	\textregistered	<code>\textregistered</code>
\ddagger	<code>\ddag</code>	\P	<code>\P</code>	\pounds	<code>\pounds</code>	$\%$	<code>\%</code>

표 3.12: AMS 짝맞춤 기호

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>
\lvert	<code>\lvert</code>	\rvert	<code>\rvert</code>	\lVert	<code>\lVert</code>	\rVert	<code>\rVert</code>

표 3.13: AMS 그리스 및 헤브루 문자

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

표 3.14: AMS 이항 관계 연산자

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

표 3.15: AMS 화살표

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

표 3.16: AMS 부정 이항 관계 연산자 및 화살표

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\nsubseteqeq
\nleqslant	\ngeqslant	\nsupseteqeq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\nleqq	\ngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\precneqq	\succneqq	\nvDash
\precnsim	\succnsim	\nVdash
\precnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\nsubseteq	\nsupseteq	\ntrianglelefteq
\subsetneqq	\supsetneqq	\ntrianglerighteq
\nleftarrow	\rightarrow	\nleftrightarrow
\nLeftarrow	\nrightarrow	\nLeftrightarrow

표 3.17: AMS 이항 연산자

$\dot{+}$	$\dot{\cdot}$	\intercal
\ltimes	\rtimes	\divideontimes
\Cup or \doublecup	\Cap or \doublecap	\smallsetminus
\veebar	\barwedge	\doublebarwedge
\boxplus	\boxminus	\circleddash
\boxtimes	\boxdot	\circledcirc
\leftthreetimes	\rightthreetimes	\circledast
\curlyvee	\curlywedge	

표 3.18: AMS 기타 기호

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

표 3.19: 수학 알파벳

보기	명령	필요한 패키지
$\mathrm{ABCDEabcde1234}$	<code>\mathrm{ABCDE abcde 1234}</code>	
$\mathit{ABCDEabcde1234}$	<code>\mathit{ABCDE abcde 1234}</code>	
$\mathnormal{ABCDEabcde1234}$	<code>\mathnormal{ABCDE abcde 1234}</code>	
\mathcal{ABCDE}	<code>\mathcal{ABCDE abcde 1234}</code>	
\mathscr{ABCDE}	<code>\mathscr{ABCDE abcde 1234}</code>	<code>mathrsfs</code>
$\mathfrak{ABCDEabcde1234}$	<code>\mathfrak{ABCDE abcde 1234}</code>	<code>amsfonts</code> or <code>amssymb</code>
$\mathbb{ABCDE\Im\Re\X\Y\Z}$	<code>\mathbb{ABCDE abcde 1234}</code>	<code>amsfonts</code> or <code>amssymb</code>

제 4 장

특별한 기능

분량이 많은 문서를 \LaTeX 으로 작성하는 경우, 찾아보기 생성이나 참고문헌 관리와 같은 특별한 기능을 활용하면 많은 도움이 된다. 이러한 \LaTeX 의 특별한 기능이나 개선점에 대해서는 *LaTeX Manual* [1]과 *The LaTeX Companion* [3]에 상세히 설명되어 있다.

제 1 절 EPS 그림 포함하기

\LaTeX 은 그림이나 사진 등을 ‘떠다니는 개체’로 처리하는 기능을 기본적으로 갖추고 있다. `figure`와 `table` 환경이 떠다니는 개체이다.

기본 \LaTeX 명령과 \LaTeX 확장 패키지만으로도 그림을 만들어내는 여러 방법이 있는데 이들 중 몇 가지는 제 5장에서 다루고 있다. 이 방식에 관심이 있다면 *The LaTeX Companion* [3]이나 *LaTeX Manual* [1]에서 더 많은 정보를 얻을 수 있다.

문서 안에 그림을 넣는 훨씬 쉬운 방법은, 그림은 전문 그래픽 소프트웨어¹로 그리고 이렇게 해서 완성된 그림을 \LaTeX 문서에 삽입하는 것이다. 외부 그림을 삽입하도록 해주는 다양한 \LaTeX 패키지들이 있다. 이 책에서는 그 중 널리 쓰이고 있고 비교적 사용하기 쉬운 캡슐화된 `POSTSCRIPT` (EPS) 형식에 대해서만 다룬다. EPS 형식의 그림을 처리하기 위해서는 `POSTSCRIPT` 프린터²가 필요하다.

D. P. Carlisle이 만든 `graphicx` 패키지가 그림 삽입을 위해 필요한 명령을 제공한다. 이 패키지는 “`graphics`”³ 패키지 묶음의 일부이다.

¹XFig, CorelDraw!, Freehand, Gnuplot, ... 등.

²`POSTSCRIPT` 출력을 얻는 다른 방법은 `GHOSTSCRIPT`을 사용하는 방법이다. 이 프로그램은 `support/ghostscript`에서 구할 수 있다. 윈도우(Windows)와 OS/2 사용자는 `GSVIEW`를 이용해서 출력할 수 있다.

³`macros/latex/graphics`

출력 장치로 POSTSCRIPT 프린터를 사용하고 `graphicx` 패키지가 설치되어 있는 경우, 그림을 문서에 포함시키려면 다음과 같은 순서를 차례로 따라하면 된다.

1. 사용하고 있는 그래픽 프로그램에서 그림을 EPS 형식으로 저장한다.⁴
2. 입력 파일의 전처리부에서 `graphicx` 패키지 사용을 선언한다.

```
\usepackage[driver]{graphicx}
```

여기서 *driver*는 “dvi를 postscript로” 변환하는 프로그램 명칭이다. 주로 사용되는 프로그램은 `dvips`이다. \TeX 자체에는 그림을 붙여넣기 위한 표준이 별도로 마련되어 있지 않기 때문에 *driver* 이름을 명시적으로 지정해야 한다. *driver*의 이름을 알아야 `graphicx` 패키지가 정확한 방법으로 `.dvi` 파일에 그림에 대한 정보를 넣을 수 있고, 프린터가 이를 해석하여 `.eps` 파일을 제대로 처리할 수 있게 되는 것이다.

3. 다음의 명령으로 *file*을 사용자 문서에 붙여넣는다.

```
\includegraphics[key=value, ...]{file}
```

옵션 인자 *key*와 여기에 할당하는 변수값인 *value*들은 쉼표로 구분하여 입력한다. *key* 값을 지정함으로써 삽입되는 그림의 폭 · 높이 · 회전값 등을 바꿀 수 있다. 표 4.1에 중요한 *key*를 정리해 두었다.

표 4.1: `graphicx` 패키지의 *key* 명칭

<code>width</code>	그림의 폭을 지정
<code>height</code>	그림의 높이를 지정
<code>angle</code>	그림을 반시계방향으로 회전
<code>scale</code>	그림의 배율을 지정

다음 예제를 보면 쉽게 이해할 수 있을 것이다.

```
\begin{figure}
\centering
```

⁴만약 그래픽 소프트웨어가 EPS 저장을 지원하지 않는다면 POSTSCRIPT 프린터 드라이버를 설치하여 EPS 파일을 얻는 방법이 있다. 예를 들면 Apple LaserWriter와 같은 드라이버를 설치한 다음, 이 드라이버를 이용하여 파일로 인쇄한다. 대개 이렇게 저장한 파일은 EPS 포맷일 것이다. EPS 파일은 한 페이지만으로 이루어진다는 점을 알아두자. 어떤 프린터 드라이버는 EPS 파일로 출력하도록 설정할 수 있다.

```
\includegraphics[angle=90, width=0.5\textwidth]{test}
\caption{A caption, explaining that this is a test.}
\end{figure}
```

이 코드는 `test.eps`라는 이름으로 저장된 그림을 문서에 삽입한다. 그림을 먼저 90도 각도로 회전하고, 그 다음에 그림의 폭이 표준 문단폭의 0.5배가 되도록 조정한다. 그림의 높이를 특별히 지정하지 않았으므로 가로×세로 비율은 원래 상태가 유지된다. 폭과 높이 값을 각각 절대값으로 지정할 수도 있음은 물론이다. 자세한 내용은 118 쪽에 있는 표 6.5를 참조하기 바란다. 이 문제에 대해 좀더 자세히 알고 싶다면 [9]와 [13]을 살펴보라.

제 2 절 참고문헌

`thebibliography` 환경을 이용하여 참고문헌 목록(bibliography)을 만들 수 있다. 각 참고문헌 항목은

```
\bibitem[label]{marker}
```

로 시작한다. 문서 본문에서 참고문헌의 단행본·논문·기사를 인용하려면 `marker`라는 명칭으로 참조한다.

```
\cite{marker}
```

```
Partl~\cite{pa} has
proposed that \ldots
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Partl [1] has proposed that ...

Bibliography

[1] H. Partl: *German T_EX*, TUGboat Volume 9, Issue 1 (1988)

옵션 인자 `label`을 명시하지 않으면 항목 번호가 자동으로 붙는다. 인자 하나를 `\begin{thebibliography}` 명령 뒤에 주는데 이것은 자동으로 붙는 항목 번호의 최대 자릿수를 지정한다. 위의 예제에서 {99}라고 한 것은 참고문헌

목록의 항목 개수가 숫자 99의 자릿수(두 자리)를 넘지 않는다는 것을 L^AT_EX에게 알려주는 것이다.⁵

큰 규모의 작업일 경우 BibT_EX 사용을 고려해 보는 것도 좋을 것이다. 대부분의 T_EX 배포판에는 BibT_EX이 포함되어 있다. BibT_EX을 이용하면 참고 문헌 데이터베이스를 관리할 수 있고, 그 데이터베이스로부터 사용자 문서에서 인용된 것들만 적절히 추출하여 참고문헌으로 제시할 수 있다. BibT_EX이 생성하는 참고문헌 목록의 외관(표시방식)은 스타일 양식(style sheets) 개념을 채택하고 있기 때문에, 널리 쓰이는 여러 가지 표준화된 디자인에 알맞은 참고문헌 목록을 쉽게 만들 수 있다.

제 3 절 찾아보기

찾아보기(index)는 책의 매우 유용한 구성요소이다. L^AT_EX과 함께 보조 프로그램 makeindex⁶를 이용하면 찾아보기를 쉽게 작성할 수 있다. 이 문서에서는 기본적인 찾아보기 목록 생성 명령에 대해서만 설명한다. 더 상세한 사항은 *The L^AT_EX Companion* [3]을 참고하라.

L^AT_EX의 찾아보기 만들기 기능이 작동하게 하려면, 문서의 전처리부(preamble)에서 makeidx 패키지를 사용하기 위해

```
\usepackage{makeidx}
```

와 같이 선언하고, 다음과 같은 특별한 찾아보기 목록 생성 명령

```
\makeindex
```

를 전처리부에 써넣어야 한다.

찾아보기 항목으로 들어갈 내용은

```
\index{key}
```

의 형식으로 작성한다. 여기서 *key*는 찾아보기 항목 표제어로 쓰이게 된다. 최종 찾아보기 목록에 표시될 참조 위치에 해당하는 본문에서 \index 명령을 사용한다. 즉 \index 명령이 쓰인 위치의 페이지 번호를 찾아보기 목록이 지시하게 된다. 표 4.2에 *key* 인자의 사용법을 설명하고 몇 가지 예를 보였다.

L^AT_EX은 입력파일을 처리하다가 \index 명령을 만나면 찾아보기 항목과 현재의 페이지 번호를, L^AT_EX 입력파일과 이름이 같고 확장명이 .idx인 특별

⁵이 인자로 숫자 이외에 *widest-label*을 지정할 수 있다. *widest-label*은 참조할 레이블이 가질 대강의 길이를 보여주는 텍스트인데, \bibitem의 옵션 인자로 주어지는 *label* 문자열 중에서 제일 긴 것을 쓰면 된다. 이렇게 하면 참고문헌 목록이나 본문의 \cite가 쓰인 위치에 숫자가 붙지 않고 *label* 문자열이 온다. —[역주]

⁶파일명에 8자 이상을 사용할 수 없는 시스템에서는 makeidx이다.

표 4.2: Index Key 사용 예제

예제	찾아보기 항목	설명
<code>\index{hello}</code>	hello, 1	기본 항목 생성
<code>\index{hello!Peter}</code>	Peter, 3	'hello'의 하위 항목
<code>\index{Sam@\textsl{Sam}}</code>	<i>Sam</i> , 2	항목 장식
<code>\index{Lin@\textbf{Lin}}</code>	Lin , 7	상동
<code>\index{Jenny textbf}</code>	Jenny, 3	페이지 번호 장식
<code>\index{Joe textit}</code>	Joe, 5	위와 같음
<code>\index{eolienne@\'eolienne}</code>	éolienne, 4	엑센트(accents) 사용

한 파일에 기록한다. 이 .idx 파일에 대하여 다음과 같이 `makeindex` 프로그램을 실행하면,

```
makeindex filename
```

`makeindex` 프로그램은 찾아보기를 정렬하여, 파일이름이 같고 확장명이 .ind 인 파일에 저장한다. L^AT_EX이 입력파일을 다시 처리할 때, 이렇게 정렬된 색인은 본문내의

```
\printindex
```

명령이 위치한 곳에 삽입된다.

L^AT_EX 2_ε 기본 패키지인 `showidx`는 찾아보기 항목을 본문의 좌측 여백에 인쇄한다. 이것은 문서의 검토와 찾아보기 확인에 매우 유용하다.

`\index` 명령은 조심스럽게 사용하지 않으면 문서의 레이아웃에 영향을 끼칠 수 있으므로 주의하여야 한다.

My Word `\index{Word}`. As opposed
to Word`\index{Word}`. Note the
position of the full stop.

My Word . As opposed to Word. Note the
position of the full stop.

제 4 절 멋진 쪽머리글

Piet van Oostrum이 작성한 `fancyhdr` 패키지⁷는 간단한 몇 개의 명령어로 문서의 면주⁸를 사용자가 설정할 수 있게 해 준다. 이 책의 페이지 상단 머리글이 바로 이 패키지를 활용하여 작성한 것이다.

⁷CTAN:tex-archive/macros/latex/contrib/fancyhdr에서 구할 수 있다.

⁸페이지의 header와 footer를 합하여 면주(面柱)라 한다. —[역주]

```

\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% with this we ensure that the chapter and section
% headings are in lowercase.
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} % delete current setting for header and footer
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % make space for the rule
\fancypagestyle{plain}{%
  \fancyhead{} % get rid of headers on plain pages
  \renewcommand{\headrulewidth}{0pt} % and the line
}

```

그림 4.1: fancyhdr 설정 예제

먼주를 사용자가 설정할 때 실제 장·절의 표제가 표시되도록 하고 싶은 경우가 있다. 이것은 약간 교묘한 방법을 써야 하는데, \LaTeX 에서는 2단계 접근 방식으로 이 문제를 해결하고 있다. 머리와 바닥 먼주를 정의할 때 현재 장·절 표제가 표시되도록 하려면 표시될 위치에 `\rightmark`와 `\leftmark` 명령을 쓴다. 이 두 명령의 값은 장절명령이 수행될 때마다 변화한다.

`\chapter` 계열 명령이 직접 `\rightmark`, `\leftmark`를 재정의하지는 않도록 하여 설정 가능성을 극대화하고 있다. 대신 별도의 명령 `\chaptermark`, `\sectionmark` 및 `\subsectionmark`를 호출하는데, 이 명령을 통하여 `\rightmark`와 `\markleft`를 간접적으로 재정의한다.

따라서, 머리 먼주에 오는 chapter 표제를 변경하고자 한다면, `\chaptermark`를 “새로 지정(renew)”하면 된다.

이 책의 먼주 설정은 `fancyhdr`를 사용한 것이다. 그 내용을 그림 4.1에 보았다. 위에 언급한 주소에서 이 패키지에 관한 문서를 구해볼 것을 권한다.

제 5 절 Verbatim 패키지

앞에서 `verbatim` 환경에 대해 알아본 적이 있다. 이 절에서는 `verbatim` 패키지에 대해 알아보도록 하자. 기본적으로 `verbatim` 패키지는 원래의 `verbatim` 환경이 갖는 몇 가지 한계를 극복할 수 있도록 `verbatim`을 재구현한 것이다.

이것 자체는 별로 대단한 것이 아니지만, `verbatim` 패키지를 구현하면서 몇 가지 기능들이 추가되었는데, 이 추가된 기능 때문에 이 패키지를 언급하는 것이다. `verbatim` 패키지는 다음 명령을 제공한다.

```
\verbatiminput{filename}
```

이 명령은 아스키 텍스트(ASCII text)를 읽어와서, `verbatim` 환경처럼 보여 준다.

`verbatim` 패키지는 ‘tools’ 패키지 묶음에 포함되어 있으므로, 대부분의 시스템에 이미 설치 되어 있을 것이다. 이 패키지에 대하여 좀더 자세히 알고 싶으면 [10]을 참조하라.

제 6 절 L^AT_EX 패키지의 다운로드와 설치

대부분의 L^AT_EX 설치 배포본은 여러 스타일 패키지를 기본적으로 설치해준다. 그러나 인터넷으로 구할 수 있는 수많은 패키지들이 더 있다. 인터넷에서 스타일 패키지를 주로 찾아볼 수 있는 곳은 CTAN(<http://www.ctan.org/>)이다.

`geometry`, `hyphenat` 또는 그밖의 여러 패키지들은 대부분 두 개의 파일로 되어 있다. 하나는 `.ins` 확장명을 가진 것이고 다른 하나는 `.dtx` 확장명을 가진 것이다. 일반적으로 패키지에 대하여 개략적인 설명을 `readme.txt`에 넣어두고 있으므로, 당연히 이 파일을 먼저 읽어야 한다.

이제 이 파일을 자신의 컴퓨터에 복사한다. 그런 다음에 (a) 자신의 T_EX 시스템이 새로운 스타일 패키지를 인식하도록 하고, (b) 관련 문서를 얻는 절차를 거친다. 첫번째 작업은 다음과 같이 한다.

1. L^AT_EX을 `.ins` 파일에 대하여 실행한다. 이렇게 하면 `.sty` 파일이 풀려 나온다.
2. 이 `.sty` 파일을 자신의 T_EX 시스템이 스타일 파일을 찾는 위치로 옮긴다. 이 위치는 일반적으로 `.../localtexmf/tex/latex` 디렉토리의 하위 디렉토리이다. (윈도우나 OS/2 사용자들은 슬래시를 백슬래시로 생각하라.⁹⁾)
3. 파일이름 데이터베이스를 갱신한다. 이 방법은 L^AT_EX 배포판에 따라 조금씩 다르다. 예를 들면 `teTeX`, `fpTeX`의 경우에는 `texhash`이고 `web2c`에서는 `mktexlsr`이다. MiKTeX은 `initexmf -update-fndb` 명령을 사용하거나 GUI 프로그램¹⁰을 이용한다.

⁹이 글에서는 UNIX 계열에서의 설치를 기준으로 설명하고 있어 디렉토리 분리자를 슬래시(/)로 표기하고 있다. 그러나 OS/2나 윈도우등은 디렉토리 경로를 백슬래시로 구분한다. —[역주]

¹⁰MiKTeX Options를 실행하거나 명령행에서 `mo`를 입력한다. —[역주]

이제 .dtx 파일에서 문서를 추출하여 보자.

1. .dtx 파일에 대하여 L^AT_EX을 실행한다. 이렇게 하면 .dvi 파일이 만들어진다. 상호 참조를 정확하게 얻기 위해서는 L^AT_EX을 몇 차례 실행해야 한다는 점에 주의하라.
2. L^AT_EX이 .idx 파일을 컴파일 중에 만들어 내었는지 확인한다. 만약 이 파일이 있으면 아래의 인덱스를 만드는 절차를 거치고 그렇지 않으면 5 단계로 넘어간다.
3. 인덱스를 얻어내기 위해서는 다음과 같이 한다.

```
makeindex -s gind.ist name
```

(*name*이 의미하는 것은 주 원본파일 이름을 확장명 없이 쓰라는 것이다.)
4. .dtx 파일에 대하여 L^AT_EX을 한 번 더 실행한다.
5. 기본적인 과정은 끝이다. 그러나 문서를 쉽게 읽으려면 .ps나 .pdf로 변환하라.

.glo(glossary) 파일이 만들어지는 경우도 있다. 이 때는 아래의 명령을 4 단계와 5 단계 사이에 실행하라.

```
makeindex -s gglo.ist -o name.gls name.glo
```

5 단계까지 가기 전에 마지막 한 번은 .dtx에 대하여 L^AT_EX을 실행해야 한다는 점을 명심하자.

제 7 절 pdfL^AT_EX 사용하기

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

PDF는 하이퍼텍스트(hypertext) 문서 형식이다. 웹 문서처럼 문서 내의 일부 단어에 하이퍼링크(hyperlink)를 걸 수 있어서 문서내의 다른 곳이나 외부 문서로 연결해준다. 하이퍼링크 설정된 단어를 클릭하면 링크된 곳으로 이동하는 것이다. L^AT_EX의 관점에서 말하자면 \ref나 \pageref가 하이퍼링크가 될 수 있다. 그리고 목차(table of content), 찾아보기(index)와 같은 구조들이 하이퍼링크들로 이루어질 수 있다는 뜻이다.

오늘날 거의 대부분의 웹 문서들은 HTML(*HyperText Markup Language*) 언어로 만든다. 그러나 이 형식은 과학 계열 문서를 작성하는 데에 중요한 결점이 두 가지 있다.

1. HTML 문서 내에서 수학적 표현이 일반적으로 지원되지 않는다. 수학적

표현의 표준이 있기는 하지만,¹¹ 대부분의 브라우저는 이 표준을 지원하지 않거나 필요한 글꼴이 빠져 있다.

2. HTML 문서를 인쇄하는 것은 가능하지만, 인쇄물은 출력환경(사용 O/S, 브라우저의 종류)이 무엇이냐에 따라 매번 결과가 다르게 나타난다. 또한 인쇄물의 품질이 \LaTeX 을 사용하여 손쉽게 얻고 있는 문서의 품질에 미치지 못한다.

\LaTeX 문서를 HTML로 바꾸는 변환기를 만들려는 시도가 여러 번 있었다. 그 중 몇 가지는 표준 \LaTeX 입력 파일로부터 읽을 만한 웹 문서를 만들어냈다고 할 만한 꽤 성공적인 것도 있다. 그러나 이 과정에서 원래 문서의 많은 부분을 희생하지 않고 결과를 얻고 있는 것은 없다. 좀더 복잡한 \LaTeX 설정과 외부 패키지를 사용하는 경우에는 여지없이 실패한다. 자신의 문서를 웹 출판하면서도 특정 타이포그래피 속성을 유지하고자 하여 PDF(*Portable Document Format*)를 고려하는 저자들이 많아졌다. 문서의 레이아웃을 유지하면서 하이퍼텍스트 연결을 가능하게 해주기 때문이다. 대부분의 요즘 브라우저들은 PDF 문서를 직접 열어 보여주는 플러그인을 갖추고 있다.

거의 모든 운영체제(O/S)에 DVI와 PS 읽기 프로그램이 갖추어져 있지만, PDF 문서를 보여주는 Acrobat Reader와 Xpdf는 더 보편적이다. 그러므로 문서를 PDF 형식으로 제공하면, 독자들이 문서에 좀더 쉽게 접근할 수 있게 될 것이다.

7.1 PDF 문서의 웹 출판

\LaTeX 원본 파일에서 PDF 파일을 만드는 것은, Hàn Thê Thành이 개발한 pdf \TeX 프로그램으로 대단히 쉽게 해낼 수 있다. \TeX 이 DVI를 만드는 것처럼 pdf \TeX 은 PDF를 만들어 낸다. \LaTeX 에 대응하는 pdf \LaTeX 도 있다.

pdf \TeX 과 pdf \LaTeX 은 te \TeX , fp \TeX , Mik \TeX , \TeX Live, CMac \TeX 등 대부분의 \TeX 배포판에서 자동으로 설치해준다.

DVI 대신 PDF를 만들기 위해서는, 명령행(command line)에서 `latex file.tex` 이라 하지 않고 `pdflatex file.tex`으로 입력하는 것으로 충분하다. 명령행 작업 환경이 아니라면 \TeX 실행 셸에서 해당하는 기능 버튼을 찾아서 누르면 된다.

\LaTeX 에서 용지 크기는 documentclass 인자로 a4paper나 letterpaper 등을 넣어 주어 정의한다. pdf \LaTeX 에서도 마찬가지이다. 그러나 pdf \TeX 은 pdf 파일의 종이 크기를 결정하기 위해서 그 실제 크기를 미리 알고 있어야 한다.

¹¹MathML이 있다. —[역주]

hyperref 패키지를 이용하면¹² 용지 크기는 자동으로 조정된다. 그렇지 않은 경우 문서의 전처리부에 다음 몇 줄을 직접 입력하여야 한다.

```
\pdfpagewidth=\paperwidth
\pdfpageheight=\paperheight
```

이어지는 절에서 일반적인 L^AT_EX과 pdfL^AT_EX의 차이점에 대하여 좀더 자세히 알아보겠다. 주요 차이점은 글꼴 사용, 포함되는 그림의 형식, 하이퍼링크 설정 등 세 가지이다.

7.2 글꼴

pdfL^AT_EX은 모든 종류의 글꼴(PK bitmaps, TrueType, POSTSCRIPT type 1...)을 사용할 수 있지만, 일반적인 L^AT_EX 글꼴인 비트맵 PK 글꼴을 사용하면, Acrobat Reader로 읽었을 때 표시되는 글꼴의 품질이 좋지 않다. 화면에서 깨끗하게 볼 수 있는 문서를 만들려면 POSTSCRIPT Type 1 글꼴만을 사용하는 것이 가장 좋다. 최근의 T_EX 배포판은 이것을 자동으로 설정하여 준다. 일단 시도하여 볼 일이고, 만약 제대로 작동한다면 이 절 전체를 건너뛰어도 무방하다.

Computer Modern과 AMSFonts 글꼴의 POSTSCRIPT Type 1 변환은 Blue Sky Research사와 Y&Y, Inc.사에서 구현하여, 글꼴 저작권을 미국 수학회에 넘겨주었다. 이 글꼴들은 1997년초 공개적으로 사용되는 것이 허용되어 현재 대부분의 T_EX 배포판에 포함되어 있다.

그렇지만, 영어 이외의 언어로 L^AT_EX 문서를 만든다면 EC, LH 혹은 CB 글꼴이 필요하다(28 쪽의 OT1 글꼴에 대한 논의를 참조하라). Vladimir Volovich는 EC/TC, EC Concrete, EC Bright, LH 글꼴 집합을 모두 포함하는 cm-super 글꼴 묶음을 만들었다. 이 글꼴 묶음은 CTAN:/tex-archive/fonts/ps-type1/cm-super에서 구할 수 있으며, T_EXLive7과 MikT_EX에 포함되어 있다. CB Greek type 1 그리스 문자 글꼴은 Apostolos Syropoulos가 비슷한 방법으로 만들었고 CTAN:/tex-archive/tex-archive/fonts/greek/cb에서 구할 수 있다. 불행히도 이 글꼴 집합 두 가지는 Blue Sky와 Y&Y사가 만든 Type1 CM 글꼴의 품질에 미치지 못한다. 자동 글꼴 보정(auto-hinting)을 이용하였기 때문에 Blue Sky/Y&Y사의 type 1 CM 글꼴을 사용한 문서보다 화면 상에서 깨끗하게 보이지 않으며, 고해상도 출력 장치에서는 원래의 비트맵 EC/LH/CB 글꼴 수준으로 출력된다.

라틴 계열 언어 문서를 만든다면, 다른 방법이 몇 가지 있다.

- aeguill 패키지(*Almost European Computer Modern with Guillemets*)를 사용해보는 방법이 있다. 문서 전처리부에 `\usepackage{aeguill}` 한 줄을 넣으면 EC 글꼴 대신 AE 가상 글꼴을 이용할 수 있다.

¹²82쪽을 참조하라

- 또다른 방법으로 mltex 패키지를 사용할 수 있다. 그러나 이것은 pdfT_EX 이 mltex 옵션으로 컴파일 된 경우에만 가능하다.¹³

MIT_EX과 AE 가상 글꼴은 CM 글꼴에서 글자를 취하여 빠진 글자들을 채워 넣고 EC 순서로 재배열하여 256문자 전체가 모두 채워진 글꼴인 것처럼 T_EX 이 이용하도록 하는 방법으로서 CM 글꼴의 고품위 type 1 포맷을 최대한 활용한다. 이 글꼴들은 T1 인코딩이므로 하이픈 처리가 라틴 계열 유럽어에도 잘 동작하게 된다. 이 방법의 유일한 결점은 가상 글꼴로 조작한 AE 문자들에 Acrobat Reader의 찾기(Find) 기능이 작동하지 않는다는 점이다. 그래서 최종 PDF 파일에서 액션트 기호 붙은 문자가 포함된 단어는 검색이 불가능하다.

러시아어에도 비슷한 C1 가상 글꼴 해결책이 있다. 이것은 <ftp://ftp.vsu.ru/pub/tex/font-packs/cifonts>에서 찾아볼 수 있다. 이 글꼴은 CTAN에서 얻을 수 있는 Bluesky 사가 제작한 표준 CM type 1 글꼴과 Paradissa와 BaKoMa의 CMCYR type 1 글꼴을 합친 것이다. Paradissa 글꼴이 러시아 문자만을 포함하고 있기 때문에 C1 글꼴에는 기타 키릴 문자가 빠져 있다.

이밖의 다른 해결책으로 CM이 아닌 일반 POSTSCRIPT type 1 글꼴을 사용하는 방법이 있다. 실제 그 중 일부는 Acrobat Reader에 포함되어 있기도 하다. 이 글꼴은 글자 크기가 다르기 때문에 문서의 페이지 레이아웃을 변경할 수 있다. 일반적으로 이 글꼴들은 공백을 매우 효율적으로 배치하는 CM 글꼴에 비해서 더 많은 공백을 차지한다. 또한 문서 전체에 걸친 시각적 일관성이 떨어질 수 있는데 그 이유는 times, helvetica, courier 글꼴(대치할 폰트로 가장 많이 사용되는 것)들이 원래 한 문서에서 동시에 사용하여 조화롭게 보이도록 디자인된 것이 아니기 때문이다.

CM이 아닌 다른 POSTSCRIPT 글꼴을 쓰려 한다면 두 가지 글꼴 모음이 사용가능하다. *palatino* 글꼴을 본문글꼴로 사용하는 pxfonts와 *times* 글꼴에 기반을 두고 있는 txfonts 패키지가 그것이다. 이들을 이용하기 위해서는 문서의 전처리부에 다음 몇 줄을 써주는 것으로 충분하다.

```
\usepackage[T1]{fontenc}
\usepackage{pxfonts}
```

주의 : 컴파일 후 .log file에 다음과 같은 내용이 들어있을 때가 있다.

```
Warning: pdftex (file eurmo10): Font eurmo10 at ... not found
```

이 에러는 문서에서 사용한다고 선언한 글꼴을 찾을 수 없었다는 뜻이다. 이 문제를 수정하지 않으면 생성된 PDF 문서를 볼 때 일부 글자가 표시되지 않는 결과를 가져온다.

지금까지 많은 사람들이 개선할 필요를 느껴온 것은, type 1 형식 CM 글꼴의 품질에 필적하는 좋은 EC 글꼴이 없다는 점이었다. 최근, 새로운 고품질의

¹³mltex에 대해서는 CTAN:/tex-archive/systems/generic/mltex을 보라. —[역주]

윌콕스 글꼴 Latin Modern(LM)이 나오으로써 좋지 않은 상황을 일소하였다. 최근 \TeX 배포판을 설치하였다면 이미 이 글꼴이 설치되어 있을 것이므로 문서의 전처리부(preamble)에 다음과 같이 적어 넣기만 하면 된다.

```
\userpackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage{textcomp}
```

이제 모든 라틴 문자 집합을 모두 지원하는 멋진 PDF 문서를 얻을 수 있다.

7.3 그림 포함하기

문서에 그림을 넣을 때 71쪽에서 설명한 `graphicx` 패키지를 쓴다. \pdfLaTeX 에서도 이 패키지를 사용하는데 이 때는 `pdftex driver` 옵션을 지정한다.

```
\usepackage[pdftex]{color,graphicx}
```

위의 예에서는 웹 문서에서 색상을 사용하여 자연스럽게 보이도록 하기 위해서 `color` 패키지 옵션을 포함하였다.

캡슐화된 POSTSCRIPT (EPS) 형식의 그림 파일은 \pdfLaTeX 에서 사용할 수 없다. `\includegraphics` 명령을 쓸 때 파일의 확장명을 지정하지 않는다면 `graphicx` 패키지는 `driver` 옵션에 따라 적절한 파일만을 찾는다. `pdftex` 옵션이 주어지면 찾는 그림 파일의 확장명은 `.png`, `.pdf`, `.jpg`, `.mps`(METAPOST)인데, 여기에 `.eps`는 들어 있지 않다.

이 문제를 벗어날 수 있는 간단한 방법은 대부분의 시스템에 갖추어져 있는 `epstopdf` 유틸리티를 이용하여 EPS 파일을 PDF 파일로 바꾸는 것이다. 벡터 그림 파일에 있어서 이러한 방식은 아주 좋은 해결책이다. 다만 비트맵 그림(사진, 스캔본)은 이러한 방법이 아주 좋은 것은 아니다. 왜냐하면, PDF 포맷은 자체적으로 PNG나 JPEG 파일을 포함할 수 있기 때문이다. PNG는 스크린샷이나 색상수가 적은 그림에 좋다. JPEG는 사진에 특히 유용하며, 파일 크기가 효율적이라는 장점이 있다.

기하학적 도형이라면 그림판에서 그리는 것보다 METAPOST와 같은 전문적인 그림그리기 언어로 그림을 기술하는 편이 훨씬 낫다. METAPOST는 대부분의 \TeX 배포판과 함께 배포되며, 상당한 양의 설명서가 같이 따라온다.

7.4 하이퍼텍스트 링크

`hyperref` 패키지는 문서내의 모든 참조 부분을 하이퍼링크로 만들어 준다. 이러한 기능이 원활하게 동작하게 하려면 `\usepackage[pdftex]{hyperref}`을 문서 전처리부의 마지막 명령으로 넣어주어야 한다.

`hyperref` 패키지의 동작을 제어하도록 설정하는 많은 옵션이 있다.

- pdftex 옵션을 지시한 다음 다른 옵션 항목을 그 뒤에 쉼표로 나누어 적어주거나

```
\usepackage[pdftex]{hyperref}
```

- \hypersetup{options} 명령으로 별도의 명령행을 구성하여도 된다.

꼭 넣어주어야 하는 옵션은 pdftex이다. 다른 옵션들은 사용자의 선택에 따라 넣을 수도 있으며, 이들은 hyperref의 동작을 기본값과는 다르게 설정할 수 있게 해준다.¹⁴ 다음 예시에서 기본값(default values)은 바로 선 글꼴로 표시하고, 기본값이 아닌 것은 기울인 글꼴로 표시한다.

bookmarks (=true, false) 문서를 볼 때 북마크 항목을 보여주거나 숨긴다.

unicode (=false, true) 비라틴계열의 언어를 Acrobat의 북마크에서 사용할 수 있게 한다.

pdftoolbar (=true, false) Acrobat의 툴바(toolbar)를 보여주거나 숨긴다.

pdfmenubar (=true, false) Acrobat의 메뉴를 보여주거나 숨긴다.

pdfwindow (=true, false) pdf가 표시될 때 초기 배율을 조정한다.

pdftitle (=text) Acrobat의 Document Info에서 보여주는 문서 제목을 정의한다.

pdfauthor (=text) PDF문서의 저자명을 정의한다.

pdfnewwindow (=true, false) 현재 문서가 아닌 다른 문서를 열 때 새창으로 열 것인지 정의한다.

colorlinks (=false, true) (false)를 넣어주면 link를 색상 박스로 표시하고, (true)를 넣어주면 link를 색상을 띤 글자로 표시한다. 이러한 link의 색상을 다음과 같은 옵션으로 조정한다. (괄호안에 기본색상을 표시하였다.)

linkcolor (=red) sections, pages 등의 내부 참조 색상

citecolor (=green) bibliography 등의 참고문헌 색상

filecolor (=magenta) 파일 link의 색상

urlcolor (=cyan) e-mail, web 등의 URL 색상

기본값을 사용할 생각이라면, 다음과 같이 한다.

¹⁴hyperref 패키지가 pdfT_EX를 위한 것만이 아님을 지적해둔다. 이 패키지는 L^AT_EX의 DVI 출력에 PDF special 코드를 넣어두었다가 dvips에 의하여 PS 파일로 변환될 때 넘겨받아서 마지막으로 Adobe Distiller로 PS를 PDF로 변환할 때 코드가 동작하도록 하는 데도 쓰인다.

```
\usepackage[pdftex]{hyperref}
```

북마크를 만들고자 하고, 링크를 색상으로 표시하려면 다음과 같이 한다.

```
\usepackage[pdftex,bookmarks,colorlinks]{hyperref}
```

PDF 파일을 인쇄용으로 만들 때는 색상 링크가 부적절하다. 색상 링크는 최종출력물에서 회색조로 표시되어 가독성이 떨어지기 때문이다. 인쇄시에 출력되지 않는 색상 상자를 이용하는 방법이 있다.

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

아예 링크를 검정색으로 설정한다.

```
\usepackage{hyperref}
\hypersetup{colorlinks,%
             citecolor=black,%
             filecolor=black,%
             linkcolor=black,%
             urlcolor=black,%
             pdftex}
```

PDF 파일의 Document Info에 필요한 정보를 넣어주려면 아래와 같이 하라.

```
\usepackage[pdftitle={Pierre Desproges},%
             pdftitle={Des femmes qui tombent},%
             pdftex]{hyperref}
```

상호참조에 자동 하이퍼링크가 되지만 특정 위치에 임의로 링크를 걸려면 다음과 같이 할 수 있다.

```
\href{url}{text}
```

이 위치에 링크가 만들어진다.

```
The \href{http://www.ctan.org}{CTAN} website.
```

라고 입력한 코드는 “CTAN”을 생성한다. “CTAN”을 클릭하면 CTAN의 웹사이트로 접속한다.

만약 링크되는 곳이, URL이 아니고 컴퓨터 내의 특정 파일이라면 \href 명령을 써서

완전한 문서는 \href{manual.pdf}{이곳}에 있습니다.

위와 같이 하면 된다. 이 명령은 “완전한 문서는 이곳에 있습니다.”라는 문자열을 만들고, “[이곳](#)”을 클릭하면, `manual.pdf` 파일을 열어 보여준다. (파일 이름은 현재 문서의 위치로부터 상대 위치로 지정한다.)

문서의 표제지에 쓰인 `\author` 명령에 `\href` 명령을 이용한 링크를 걸어서 독자가 저자 자신에게 쉽게 이메일을 보낼 수 있게할 수 있다.

```
\author{Mary Oetiker <\href{mailto:mary@oetiker.ch}%
{mary@oetiker.ch}>}%
```

여기서 이메일 주소를 링크로만 넣지 않고 페이지 자체에도 나타나게 하였다. `\href{mailto:mary@oetiker.ch}{Mary Oetiker}` 라고 링크를 걸면 Acrobat에서는 잘 동작하지만, 인쇄했을 때 이메일 주소가 나오지 않기 때문이다.

7.5 링크 관련 문제

다음과 같은 메시지를 만날 때가 있다.

```
! pdfTeX warning (ext4): destination with the same identifier
(name{page.1}) has been already used, duplicate ignored
```

이것은 카운터가 재설정될 때 나타난다. 예를 들자면, `book` 클래스에서 제공하는 `\mainmatter` 명령을 이용하는 경우 페이지 번호 카운터를 첫번째 장(chapter) 앞에서 1로 설정한다. 그러나, 책의 앞부분 서문(preface)도 페이지 번호가 1이기 때문에, 문서 내에서 “page 1”을 가리키는 링크가 둘 이상이 되어 “중복이 무시”되는 일이 발생하였다.

카운터를 별도로 세도록 하려면 `hyperref`의 옵션으로 `plainpages=false`를 넣어준다. 그러나 이 방법도 페이지 카운터에만 도움이 된다. 이보다 훨씬 과격한 방법은 `hypertex=names=false` 옵션을 쓰는 것이지만 이번에는 색인(index)의 페이지 참조가 하이퍼링크로 작동하지 않는다.

7.6 책갈피 관련 문제

책갈피(bookmark) 모양이 저자가 예상한 것과 항상 동일한 것은 아니다. 책갈피는 “단순 문자열”일 뿐이므로 사용할 수 있는 문자가 훨씬 제한되며 일반적인 L^AT_EX 문서에서처럼 다양하지 못하다. `hyperref`는 이러한 문제를 감지하면 다음과 같은 경고 메시지를 보여준다.

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

이 문제를 해결하는 방법은 책갈피에 들어가기에 적합하지 않은 텍스트 대신 넣을 수 있는 책갈피용 텍스트를 제공해주는 것이다.

```
\texorpdfstring{TeX text}{Bookmark Text}
```

수식의 경우가 이런 방법이 쓰여야 할 가장 좋은 예이다.

```
\section{\texorpdfstring{$E=mc^2$}%
{E\ =\ mc\texttt{two}superior}}
```

여기서 `\section{$E=mc^2$}`를 책갈피에서는 “E=mc²”로 바꾸어 표시하도록 하였다.

색상 변경도 책갈피에서는 적합하지 않다.

```
\section{\textcolor{red}{Red !}}
```

위의 코드는 책갈피에서 “redRed!”와 같이 나타난다. `\textcolor` 명령은 무시되지만 (red)라는 인자는 출력된다.

이 경우 다음과 같이 지정하여

```
\section{\texorpdfstring{\textcolor{red}{Red !}}{Red\ !}}
```

책갈피용 문자열을 별도로 제공하면 훨씬 나은 결과를 얻는다.

7.6.1 동일한 원본을 L^AT_EX과 pdfL^AT_EX으로 처리하기

이론적으로 말하자면 동일한 원본 문서가 L^AT_EX과 pdfL^AT_EX 양쪽 모두에서 잘 컴파일 되어야 한다. 그러나 현실적으로 가장 큰 문제점은 그림 파일 처리이다. 손쉬운 해결 방법은 `\includegraphics` 명령에서 그림 파일을 지시할 때 확장명을 의도적으로 명시하지 않는 것이다. 그러면 현재 디렉토리에서 적절한 형식의 그림 파일을 자동적으로 찾을 것이다. 사용자가 하여야 할 일은, 그림 파일들을 미리 적절한 형태로 만들어 두는 것이다. L^AT_EX은 .eps 파일을 찾고, pdfL^AT_EX은 .png, .pdf, .jpg, .mps 파일을 찾는다.

문서를 PDF로 만들 때 별도의 코드를 사용하고 싶다면 `ifpdf`¹⁵ 패키지를 문서의 전처리부에 넣어준다. 이 패키지가 이미 설치되어 있기가 쉽다. 그렇지 않더라도 MiK_TE_X의 경우 이 패키지를 처음 사용할 때 자동으로 설치될 것이다. 이 패키지는 조건별 분기 코드를 쉽게 만들 수 있는 `\ifpdf`라는 명령을 정의하고 있다. 다음 예제는 인쇄 단가를 낮추기 위하여 인쇄물은 흑백 PostScript로 만들면서 컴퓨터상에서 보기 위한 PDF는 색상이 있도록 만드는 경우이다.

¹⁵이 패키지를 사용하여야 하는 이유를 더 알고 싶다면, T_EX FAQ의 <http://www.tex.ac.uk/cgi-bin/texfaq2html?label=ifpdf> 항목을 읽어 보라.

```

\ifpdf
  \usepackage[T1]{fontenc}
  \usepackage{aeuill}
  \usepackage[pdftex]{graphicx,color}
  \usepackage[pdftex,colorlinks]{hyperref}
\else
  \usepackage[T1]{fontenc}
  \usepackage[dvips]{graphicx}
  \usepackage[dvips]{hyperref}
\fi

```

위의 예제에서, PDF를 만들지 않는 경우에도 `hyperref` 패키지를 포함하였는데 이것은 `\href` 명령에서 에러가 발생하지 않게 하기 위해서이다. 이렇게 하지 않는다면 `\href` 명령이 나올 때마다 그것을 처리하는 별도의 코드를 작성해야 하기 때문에 번거롭다.

최근의 T_EXLive와 같은 배포판에서는, 설정 파일 `graphics.cfg`과 `color.cfg`의 설정에 의해 `pdftex`과 `dvips`가 실행될 때 각각 자동으로 `graphicx`와 `color` 패키지를 부르도록 되어 있다는 사실을 알아두자.

제 8 절 DVIPDFM_x로 PDF 만들기¹⁶

By KTUG lshort translation project team <info@mail.ktug.or.kr>

pdfL^AT_EX이 매우 훌륭한 프로그램이지만 한글 사용 환경에서 pdf를 제작하려 할 때는 몇 가지 문제점이 있다.

1. pdfL^AT_EX으로는 한글 텍스트의 검색·추출이 가능한 pdf 파일을 제작할 수가 없다.
2. 한글 트루타입 글꼴을 사용하는 경우 .ttc 글꼴을 직접 포함(embed)할 수 없고 기욤인 글꼴이나 굵은 글꼴을 표현하기 어렵다.
3. 적어도 H^AL^AT_EX을 사용하는 한 한글 pdf 책갈피를 구현하는 것이 (불가능하지는 않지만) 쉽지 않다.

이 가운데 세번째 것은 외부 유틸리티를 이용하여 책갈피 파일을 유니코드로 변환해주는 번거로운 절차를 거쳐야 한다. 그러나 dhucs를 사용한다면 pdfL^AT_EX에서도 pdf 책갈피를 손쉽게 만들 수 있다. 그러므로 이 문제는 어느 정도 해결된 것으로 보아도 될 것이다.¹⁷ 첫번째 문제와 두번째 문제는 현재

¹⁶이 절은 이 문서의 한국어 번역자가 추가하였다.

¹⁷H^AL^AT_EX 1.0 이전 버전에서는 hfont가 아닌 한글화 서식 스타일인 hangul을 사용하면 `hyperref` 패키지를 쓸 수 없었다. 새 버전에서 충돌 문제는 어느 정도(충돌을 해결하는 스타일을 별도로 없어야 한다) 생기지 않게 되었으나 여전히 책갈피와 같은 pdf 특유의 기능을 구현하는 데는 한계가 있는 것으로 보인다.

로서 pdf_TE_X의 개선을 기다려보아야 할 것이다.¹⁸

그래서 한글 문서를 pdf로 만들 때 DVIPDFM_x를 이용한 방법을 권장하고 있다. 이 프로그램은 Mark Wicks의 dvipdfm을 조진환(ChoF)·히라타 두 분이 확장한 것이다. 이 프로그램에 대한 자세한 사항은 <http://project.ktug.or.kr/dvipdfmx>를 방문해보기 바란다. 현재 MiK_TE_X과 W32_TE_X에는 이 프로그램이 포함되어 배포되고 있으며 MiK_TE_X 사용자를 위한 실행 프로그램을 KTUG의 ftp를 통하여 구할 수 있다. 그밖의 배포판 사용자라면 소스를 내려 받아서 직접 컴파일·빌드할 수 있다. DVIPDFM_x는 동아시아어 관련 트루타입 처리 능력 이외에도 Metapost 지원, ConTeXt 지원, PDF 암호화 기능 등 여러 가지 기능이 추가되어 가고 있는 프로그램이다.

DVIPDFM_x는 pdf 제작을 위한 2단계 접근을 취하고 있다. 원본 파일로부터 dvi 없이 pdf를 바로 만들어내는 pdf_TE_X과는 달리 우선 L_AT_EX에 의하여 dvi 출력을 얻고 그 다음에 이 dvi를 pdf로 변환하는 것이다.¹⁹ 그러므로 다음과 같은 단계를 거쳐서 pdf가 만들어진다.

```
#> latex foo
#> dvipdfmx foo
```

8.1 한글 pdf 책갈피

pdf 파일에 한글 책갈피를 넣으려면 DVIPDFM_x가 나중에 처리할 수 있도록 `\special` 코드를 미리 넣어두어야 한다. dhucs의 경우 이것은 다음과 같이 한다.

```
\usepackage{dhucs-ucshyper}
```

이 행으로 hyperref이 함께 로드되기 때문에 별도로 hyperref을 올리는 명령을 주면 안 된다. hyperref 설정은 `\hypersetup` 명령을 이 뒤에 바로 이어써서 할 수 있다.

또는 DVIPDFM_x의 tounicode 기능을 이용하여 다음과 같이 하는 방법도 있다. 이를 위해서는 UTF8-UCS2라는 이름의 CMap 파일을 미리 구하여 T_EX 시스템이 CMap을 찾는 위치²⁰에 넣어두고 texhash해두어야 한다.

```
\usepackage[dvipdfm,...,CJKbookmarks,...]{hyperref}
\AtBeginDvi{\special{pdf: tounicode UTF8-UCS2}}
```

¹⁸알려진 바에 따르면 pdf_TE_X 자체가 한글의 경우와 관계가 깊은 트루타입 글꼴 지원과 subfont 기능을 강화하는 준비가 진행되고 있다고 한다. 그러나 이 글을 쓰는 현재 subfont 기능을 갖춘 pdf_TE_X 베타판은 나와 있으나 한글 텍스트의 검색과 추출 등을 완전히 구현하는 데는 얼마간의 시간이 더 필요할 듯하다.

¹⁹pdf를 얻는 ‘3단계 접근’은 우선 latex으로 dvi를 얻고 이 dvi를 dvips로 ps 변환한 다음 Adobe Distiller나 GhostScript의 ps2pdf를 이용하여 다시 pdf로 변환하는 것이다.

²⁰te_TE_X 3.0이라면 fonts/cmap 아래이다. —[역주]

이 코드의 첫 줄에서 `hyperref` 패키지를 로드할 때 주의해야 할 점이 명시되어 있다. 반드시 `dvipdfm(dvipdfmx가 아님에 주의하라)` 옵션을 지정하고 한글 채갈피를 위해서 `CJKbookmarks` 옵션을 `bookmarks` 옵션 대신 지시한다. 그 다음 줄은 dvi에 넘겨주는 pdf special 코드이다. `tounicode`라는 pdf 명령을 포함하도록 하고 있는 부분이다. 이 코드에 의하여 적당한 CMap 파일을 찾아서 pdf를 만들 때 적용한다.

8.2 그림 삽입

그림을 처리할 때 DVIPDFM_x가 편리한 점은 .eps 그림을 외부 그림 변환기를 이용하여 자동으로 .pdf로 변환하여 포함하여 준다는 점이다. 그러나 아무래도 외부 유틸리티를 실행하는 데 시간이 걸릴 수밖에 없기 때문에 pdfL^AT_EX의 경우처럼 처음부터 .pdf나 .png, .jpg 그림을 준비해두는 편이 좋다. 외부 그림을 불러들이기 위한 설정은 다음과 같이 한다.

```
\usepackage[dvipdfm]{graphicx}
```

안타깝게도 `graphicx` 패키지의 `dvipdfm` 드라이버 설정 파일은 그다지 효율적이지 못하다. 위에 소개한 DVIPDFM_x 웹사이트에서 `dvipdfmx` 드라이버 설정 파일을 받을 수 있으므로 이를 이용하는 방법도 있지만 간단히 다음과 같이 설정하여도 괜찮다.

```
\usepackage{graphicx}
\DeclareGraphicsExtensions{.jpg,.pdf}
\DeclareGraphicsRule{.jpg}{eps}{.bb}{}
```

다만 주의할 것은 jpg, png, pdf의 경우 미리 그림을 준비할 때 .bb라는 바운딩 박스(bounding box) 정보 파일을 따로 준비해두어야 한다는 점이다. 이를 위하여 `ebb`라는 유틸리티를 사용한다.

```
#> ebb foo.pdf
```

이것은 `foo.pdf` 그림 파일로부터 `foo.bb` 파일을 만들어낸다.

8.3 한글 글꼴 기타

한글 글꼴로는 POSTSCRIPT 글꼴인 UHC 글꼴과²¹ 트루타입 글꼴을 사용할 수 있다. 트루타입을 이용할 때만 텍스트 검색 추출이 가능한 pdf를 만들 수 있음에 주의하라. 트루타입으로 일반적인 것은 UHC POSTSCRIPT 글꼴을 트루타입 변환한 은글꼴²²이 있다. 이 글꼴들은 GPL 라이선스로 배포되는 한글

²¹ H^AL^AT_EX의 표준 글꼴이다.

²² <http://faq.ktug.or.kr/faq/>의 은글꼴 페이지와 <http://kldp.net/projects/unfonts>를 참고하라.

글꼴 묶음이다. 이밖에 한글 글꼴 사용에 관한 더 상세한 사항은 KTUG(<http://www.ktug.or.kr>)의 이곳저곳을 참고하기 바란다.

DVIPDFM_x 사용은 한글 문제를 해결해주지만 처음부터 pdfT_EX을 겨냥하여 만들어진 몇 가지 패키지들에 대해서 완전한 호환성을 보장해주지는 않는다는 것을 지적해두는 것이 공평할 것이다. 예컨대 제 9절에서 소개하는 beamer 패키지는 pdfL_AT_EX을 사용할 때 가장 좋은 결과를 얻을 수 있으며 pstricks와 같은 POSTSCRIPT를 직접 사용하는 패키지는 DVIPDFM_x와 함께 사용하기 위해서 약간의 트릭을 동원하여야 한다.²³ pstricks를 사용하는 prosper와 같은 패키지도 같은 문제가 있다.

제 9절 beamer class를 이용한 발표문서 작성

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

연구 결과를 발표하려면 칠판이나 투명용지(OHP)를 이용하는 방법이 있으며, 휴대용 컴퓨터로부터 프레젠테이션 소프트웨어를 이용하여 직접 보여줄 수도 있다.

beamer 클래스를 pdfL_AT_EX와 함께 사용하여 발표 자료를 PDF로 만들 수 있다. 형편이 좋은 사람이 사용하는 PowerPoint를 이용하는 경우와 비교해도 거의 차이가 없는 자료를 만들 수 있으며, 게다가 Acrobat Reader가 어떤 시스템에서든 사용 가능하기 때문에 훨씬 이식성이 높다.

beamer 클래스는 화면 프레젠테이션에 적합한 옵션을 적용한 graphicx, color, hyperref 패키지를 사용한다.

그림 4.2에서 보인 코드를 PDFL_AT_EX으로 컴파일하면, 두 페이지 PDF 파일을 얻게 된다. 첫 페이지는 표지이고 두번째 페이지에는 발표 과정에서 한 단계씩 보여주는 몇 개의 항목이 들어 있다.

prosper는 PostScript 처리 단계를 거친다. 그리고 ppower4 패키지로 발표 자료를 만들 때는 후처리 과정이 필요하다. 그러나 beamer 클래스는 즉시 사용 가능한 PDF 파일을 생성한다는 것이 장점이다.

beamer 클래스를 사용하면, 동일한 원본 파일로부터 여러 형태의 문서를 생성할 수 있다. 입력 파일에 꺾쇠표(<, >)로 지정된 각각의 문서 형태별 지시사항에 관한 특별한 코드들을 포함하고 있으면 된다. 사용 가능한 문서 모드는 다음과 같다.

beamer 위에서 설명한 PDF 발표자료용

trans 슬라이드용

²³pstricks의 문제는 DVIPDFM_x만의 문제가 아니라 pdfT_EX에 대해서도 마찬가지이다. 이 패키지는 dvips를 이용하는 이른바 '3단계 접근'만이 거의 유일한 직접적인 사용방법이다. 여기서 '트릭'이라 한 것은 pdftricks라는 패키지를 통하여 pdfL_AT_EX이나 DVIPDFM_x에서 pstricks 코드를 간접적으로 이용하는 방법이 있음을 지적인 것이다.


```

\documentclass[10pt]{beamer}
\mode<beamer>{%
  \usetheme[hideothersubsections,right,width=22mm]{Goettingen}
}

\title{Simple Presentation}
\author[D. Flipo]{Daniel Flipo}
\institute{U.S.T.L. \& GUTenberg}
\titlegraphic{\includegraphics[width=20mm]{USTL}}
\date{2005}

\begin{document}

\begin{frame}<handout:0>
  \titlepage
\end{frame}

\section{An Example}

\begin{frame}
  \frametitle{Things to do on a Sunday Afternoon}
  \begin{block}{One could \ldots}
    \begin{itemize}
      \item walk the dog\ldots \pause
      \item read a book\pause
      \item confuse a cat\pause
    \end{itemize}
  \end{block}
  and many other things
\end{frame}
\end{document}

```

그림 4.2: beamer 클래스 샘플 코드

handout 발표자료 인쇄용

기본값은 `beamer`이며, 클래스 선택사항 설정으로 변경할 수 있다. 예를 들어 발표자료 인쇄물을 만들려면 `\documentclass[10pt,handout]{beamer}`와 같이 정의한다.

테마를 선택하여 화면 프레젠테이션의 모양을 바꿀 수 있다. `beamer` 클래스와 함께 제공되는 테마 중에서 고를 수도 있고, 직접 만들 수도 있다. 테마에 대한 자세한 정보는 `beamer` 클래스 문서 `beameruserguide.pdf`를 참조하라.

그림 4.2에 있는 코드를 좀더 자세히 살펴보자.

`\mode<beamer>` 발표자료의 화면용 버전으로 목차가 들어 있는 바로가기 패널(navigation panel)을 보여주기 위하여 *Goettingen* 테마를 선택하였다. 옵션으로 바로가기 패널의 크기(여기서는 22mm)와 위치(본문의 오른쪽)를 선택할 수 있다. *hideothersubsections* 옵션은 다른 소절은 보여주지 않고 장(chapter)의 표제와 현재 위치 소절 표제(subsection)만을 표시한다. `\mode<trans>` 와 `\mode<handout>`에는 특별한 설정값은 없고 기본 레이아웃으로 표시한다.

`\title{}`, `\author{}`, `\institute{}`, `\titlegraphic{}` 명령은 표제면의 내용을 정의한다. `\title[]{}` 과 `\author[]{}` 에 주어진 옵션 인자는 *Goettingen* 테마에서 나타나는 패널에 표시될 별도의 제목과 저자명으로 쓰인다.

패널 상의 제목과 부제목은 일반적인 `\section{}`, `\subsection{}` 명령으로 만드는데 이 명령들은 `frame` 환경의 바깥에 두어야 한다.

화면 아래쪽 작은 바로가기용 아이콘은 문서의 특정 위치를 바로 찾아가도록 해준다. 이 아이콘은 사용자가 선택한 테마와 무관하다.

각 페이지의 내용은 `frame` 환경 안에 기술한다. 꺾쇠표(<, >)안에 인자를 기술할 수 있는데 이것은 프레젠테이션의 버전에 따라 특정 프레임이 나타나지 않도록 한다. 위의 예에서는 `<handout:0>` 인자 때문에 발표자료 인쇄물(handout) 출력시 첫페이지가 출력되지 않는다.

각 슬라이드마다 표제면 슬라이드의 제목과 다른 별도의 제목을 붙이는 것이 좋다. 이것은 `\frametitle{}` 명령으로 설정한다. 부제(subtitle)가 필요하다면, 예제에서와 같이 `block` 환경을 이용한다. 장절명령 `\section{}`, `\subsection{}`은 슬라이드 상에서 제대로 된 결과를 내지 않는다는 점에 주의하라.

`itemize` 환경에서 `\pause` 명령은 항목을 하나하나 차례로 보여준다. `\only`, `\uncover`, `\alt`, `\temporal`과 같은 그밖의 프레젠테이션 효과도 확인해보기 바란다. 나중에 발표자료를 조정하기 위하여 꺾쇠표(<, >)를 사용할 수도 있을 것이다.

어떠한 경우든지, beamer 클래스 설명서 `beameruserguide.pdf`를 꼭 통독하여 이 클래스가 어떤 기능을 갖추고 있는지 확인해보도록 하라. 이 패키지는 활발하게 개발되는 중이므로, 최신의 소식을 얻으려면 beamer 클래스의 홈페이지인 <http://latex-beamer.sourceforge.net/>을 방문해보기 바란다.

제 5 장

수학 관련 그림 그리기

대부분 사람들은 텍스트를 조판하기 위해서 \LaTeX 을 쓴다. 그러나 내용과 구조를 중시하지 않는 접근 방법이 간편할 때도 있다. 제한이 있기는 하지만, \LaTeX 도 그림을 텍스트로 묘사해서 출력하는 방법을 한 가지 제공하고 있다. 또한 이 제한을 극복할 수 있는 많은 \LaTeX 확장 패키지들도 있다. 이 절에서는 그 가운데 몇 가지를 살펴본다.

제 1 절 개관

`picture` 환경은 \LaTeX 내부에서 직접 그림을 프로그래밍할 수 있도록 해준다. 자세한 설명은 *\LaTeX Manual* [1]에 나와 있다. 하지만 이 환경은 선분의 기울기나 원의 지름을 선택할 수 있는 폭이 제한되어 있는 것과 같은 심각한 한계가 있다. 한편, \LaTeX 의 `picture` 환경에는 `\qbezier` 명령이 있는데 여기서 “q”란 “2차식(quadratic)”을 뜻한다. 수학 계산을 조금 하여야 하지만 우리가 자주 쓰는 원·타원·포물선과 같은 곡선들은 2차 베지어 곡선(quadratic Bézier curve)을 이용하면 만족스러운 정도로 비슷하게 그릴 수 있다. 나아가, Java와 같은 프로그래밍 언어를 사용하여 \LaTeX 입력 파일의 `\qbezier` 부분을 만든다면 `picture` 환경은 훨씬 강력해질 것이다.

\LaTeX 에 직접 프로그래밍하여 그림을 그리는 것이 아주 한정된 경우에만 가능하고 종종 귀찮기도 하지만, 그래도 이런 일을 할 만한 값어치는 있다. 이렇게 만들어진 문서는 “작은” 바이트 크기를 갖는 파일이 되고, 같이 따라 다녀야 할 추가 그림 파일도 없기 때문이다.

`epic`과 `eepic` (*The \LaTeX Companion* [3]에 설명되어 있다) 패키지나 `pstricks`는 원래의 `picture` 환경이 지닌 제한들을 없애고 \LaTeX 의 그림 그리기 기능을 더욱 강력하게 해준다.

앞의 두 패키지는 `picture` 환경을 확장하였을 뿐이지만 `pstricks` 패키지는

그 자체가 `pspicture`라는 그림 그리기 환경을 제공한다. `pstricks`가 가진 강력함은 이 패키지가 `POSTSCRIPT` 언어의 능력을 그대로 이용한다는 사실에서 나온다. 여기서 더 나아가, 특정한 목적에 알맞은 수많은 패키지들도 나와 있다. 이러한 여러 가지 패키지들에 대해서는 *The L^AT_EX Graphics Companion* [4](*The L^AT_EX Companion* [3]과 혼동하지 말 것)에 자세히 설명되어 있다.

L^AT_EX와 관련된 가장 강력한 그림 그리기 도구는 아마 METAPOST일 것이다. 이것은 도널드 크누쓰의 METAFONT 프로그램과 쌍둥이다. METAPOST는 METAFONT라는 매우 강력하고 수학적으로 복잡한 프로그래밍 언어의 특징을 함께 가지고 있다. METAPOST가 METAFONT와 다른 점은 그림을 비트맵이 아니라 캡슐화된 `POSTSCRIPT` 파일(EPS)을 만들어 낸다는 점이다. 이렇게 출력된 파일은 그대로 L^AT_EX으로 가져올 수 있다. 이 프로그램에 대한 입문 자료로는 *A User's Manual for MetaPost* [15]나 [17]의 안내서를 참고하도록 한다.

L^AT_EX과 T_EX이 그림(과 글꼴)을 다루는 방법에 관한 좀 더 자세한 내용들은 *T_EX Unbound* [16]에 나와 있다.

제 2 절 picture 환경

By Urs Oswald <osurs@bluewin.ch>

2.1 기본 명령

`picture` 환경¹은 명령어 하나 또는 두 개로 만들어진다.

```
\begin{picture}(x,y)...\end{picture}
```

또는

```
\begin{picture}(x,y)(x_0,y_0)...\end{picture}
```

숫자 x , y , x_0 , y_0 은 `\unitlength`를 가리킨다. 이 값은 언제라도 다음과 같이 재설정할 수 있지만 `picture` 환경 안에서는 바꿀 수 없다.

```
\setlength{\unitlength}{1.2cm}
```

`\unitlength`의 기본값은 1pt이다. 첫번째 순서쌍 (x, y) 는 그림에 사용될 사각형 영역을 문서 안에서 확보하는 기능을 한다. 두번째 순서쌍 (x_0, y_0) 은

¹미거나 말거나지만, `picture` 환경은 박스 밖에서도 작동한다. 표준 L^AT_EX 2_ε에서 아무런 패키지도 로드할 필요가 없다.

필요할 때만 활용하면 되는데, 확보한 사각형 영역의 왼쪽 아래편 모서리에 좌표값을 지정하는 것이다.

그리기 명령은 대부분 두 가지 형태 가운데 하나다.

```
\put(x,y){object}
```

또는

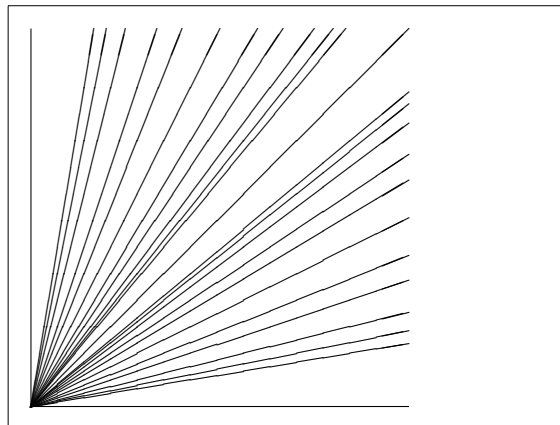
```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

Bézier 곡선은 예외이다. 이 곡선을 그리려면 다음과 같이 한다.

```
\qbezier(x_1,y_1)(x_2,y_2)(x_3,y_3)
```

2.2 선분

```
\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
  \put(0,0){\line(0,1){1}}
  \put(0,0){\line(1,0){1}}
  \put(0,0){\line(1,1){1}}
  \put(0,0){\line(1,2){.5}}
  \put(0,0){\line(1,3){.3333}}
  \put(0,0){\line(1,4){.25}}
  \put(0,0){\line(1,5){.2}}
  \put(0,0){\line(1,6){.1667}}
  \put(0,0){\line(2,1){1}}
  \put(0,0){\line(2,3){.6667}}
  \put(0,0){\line(2,5){.4}}
  \put(0,0){\line(3,1){1}}
  \put(0,0){\line(3,2){1}}
  \put(0,0){\line(3,4){.75}}
  \put(0,0){\line(3,5){.6}}
  \put(0,0){\line(4,1){1}}
  \put(0,0){\line(4,3){1}}
  \put(0,0){\line(4,5){.8}}
  \put(0,0){\line(5,1){1}}
  \put(0,0){\line(5,2){1}}
  \put(0,0){\line(5,3){1}}
  \put(0,0){\line(5,4){1}}
  \put(0,0){\line(5,6){.8333}}
  \put(0,0){\line(6,1){1}}
  \put(0,0){\line(6,5){1}}
\end{picture}
```



선분을 그리는 명령은 다음과 같다.

```
\put(x,y){\line(x1,y1){length}}
```

\line 명령에는 인자가 두 개 있는데, 각각

1. 방향 벡터,
2. 길이

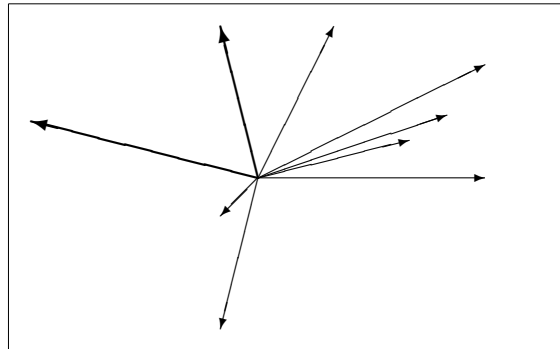
를 나타낸다. 방향 벡터는 다음 정수값으로만 줄 수 있다.

$$-6, -5, \dots, 5, 6,$$

그리고 두 값은 서로소(1 말고는 약수를 가지지 않는)이어야 한다. 그림은 제1 사분면에서 그릴 수 있는 25가지 기울기를 보여준다. 길이값은 \unitlength 에 비례한다. 길이 인자는 세로 선분에서는 수직 좌표이고 그밖의 모든 경우에는 수평 좌표이다.

2.3 화살표

```
\setlength{\unitlength}{1mm}
\begin{picture}(60,40)
  \put(30,20){\vector(1,0){30}}
  \put(30,20){\vector(4,1){20}}
  \put(30,20){\vector(3,1){25}}
  \put(30,20){\vector(2,1){30}}
  \put(30,20){\vector(1,2){10}}
  \thicklines
  \put(30,20){\vector(-4,1){30}}
  \put(30,20){\vector(-1,4){5}}
  \thinlines
  \put(30,20){\vector(-1,-1){5}}
  \put(30,20){\vector(-1,-4){5}}
\end{picture}
```



화살표를 그리는 명령은 다음과 같다.

```
\put(x,y){\vector(x1,y1){length}}
```

화살표는 선분에 비해 쓸 수 있는 방향 벡터 인수가 더 적어서 다음 정수만을 사용할 수 있다.

$$-4, -3, \dots, 3, 4.$$

이 때 인수들은 마찬가지로 서로소이어야 한다. 왼쪽 위를 향하는 화살표 두 개에 적용된 \thicklines 명령이 가져오는 효과를 주의해서 보자.

2.4 원

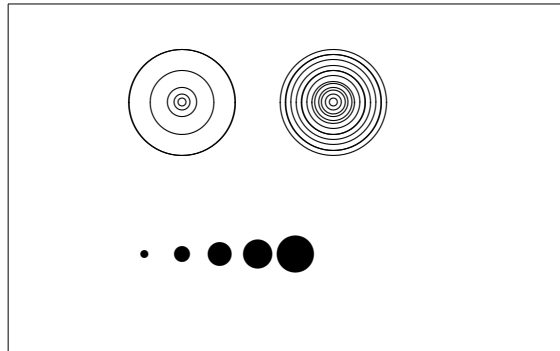
```

\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
  \put(20,30){\circle{1}}
  \put(20,30){\circle{2}}
  \put(20,30){\circle{4}}
  \put(20,30){\circle{8}}
  \put(20,30){\circle{16}}
  \put(20,30){\circle{32}}

  \put(40,30){\circle{1}}
  \put(40,30){\circle{2}}
  \put(40,30){\circle{3}}
  \put(40,30){\circle{4}}
  \put(40,30){\circle{5}}
  \put(40,30){\circle{6}}
  \put(40,30){\circle{7}}
  \put(40,30){\circle{8}}
  \put(40,30){\circle{9}}
  \put(40,30){\circle{10}}
  \put(40,30){\circle{11}}
  \put(40,30){\circle{12}}
  \put(40,30){\circle{13}}
  \put(40,30){\circle{14}}

  \put(15,10){\circle*{1}}
  \put(20,10){\circle*{2}}
  \put(25,10){\circle*{3}}
  \put(30,10){\circle*{4}}
  \put(35,10){\circle*{5}}
\end{picture}

```



원을 그리는 명령은 다음과 같다.

`\put(x , y){\circle{ $diameter$ }}`

여기서 인자 (x, y) 는 원의 중심이고 $diameter$ 는 원의 지름(반지름이 아님)이다. `picture` 환경으로 그릴 수 있는 가장 큰 원은 지름 약 14mm 정도까지만다. 또한 지름이 이보다 작은 값이라고 해서 모든 지름의 원을 다 그릴 수 있는 것도 아니다. `\circle*` 명령은 가운데가 채워진 원을 그린다.

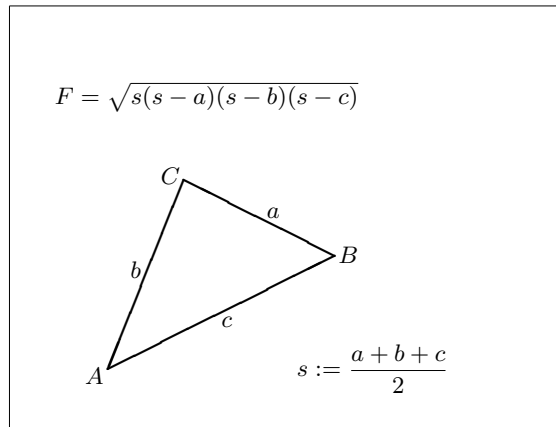
선분과 마찬가지로, `eepic`나 `pstricks`와 같은 추가 패키지를 사용해야 할 수도 있다. 이 패키지들에 대한 설명은 *The L^AT_EX Graphics Companion* [4]을 보도록 한다.

`picture` 환경 자체를 이용할 수도 있다. 약간 귀찮은 계산을 기꺼이 할 수 있다면 (또는 프로그램을 이용해서 계산을 할 수 있다면) 몇 개의 2차 베

지어 곡선을 합쳐서 원이나 타원을 원하는 대로 그릴 수 있다. *Graphics in L^AT_EX 2_ε* [17]을 보면 이에 대한 보기와 Java 소스가 있다.

2.5 텍스트와 수식

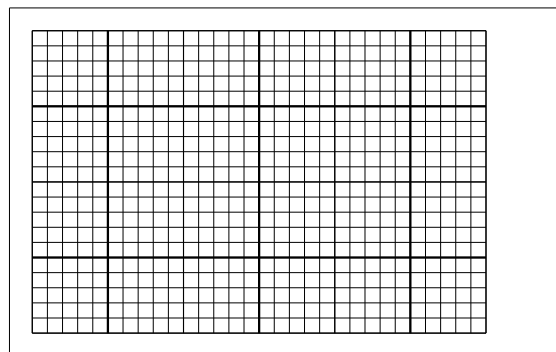
```
\setlength{\unitlength}{1cm}
\begin{picture}(6,5)
  \thicklines
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.7,0.3){$A$}
  \put(4.05,1.9){$B$}
  \put(1.7,2.95){$C$}
  \put(3.1,2.5){$a$}
  \put(1.3,1.7){$b$}
  \put(2.5,1.05){$c$}
  \put(0.3,4){$F=$
    \sqrt{s(s-a)(s-b)(s-c)}$}
  \put(3.5,0.4){$\displaystyle
    s:=\frac{a+b+c}{2}$}
\end{picture}
```



이 보기는 `picture` 환경 안에서 `\put` 명령을 사용하여 글과 수식을 일반적인 방법으로 써넣을 수 있다는 것을 보여준다.

2.6 \multiput과 \linethickness 명령

```
\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){31}%
    {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
    {\line(1,0){30}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){7}%
    {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
    {\line(1,0){30}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){3}%
    {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
    {\line(1,0){30}}
\end{picture}
```

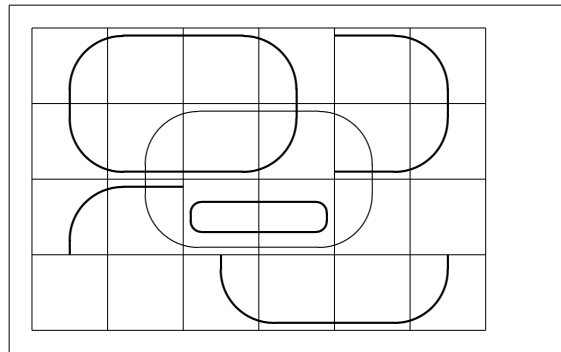



```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

이 명령은 출발점, 한 개체로부터 다음 개체까지의 진행 벡터, 개체의 수, 그려질 개체 코드, 이렇게 네 개의 인자를 갖는다. `\linethickness` 명령은 수평선과 수직선에만 효과가 있고 사선 선분이나 원에는 적용하지 않는다. 그러나 2차 베지어 곡선에는 효과가 나타난다.

2.7 알모양 곡선. `\thinlines`와 `\thicklines` 명령

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}%
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}%
    {\line(1,0){6}}
  \thicklines
  \put(2,3){\oval(3,1.8)}
  \thinlines
  \put(3,2){\oval(3,1.8)}
  \thicklines
  \put(2,1){\oval(3,1.8)[t]}
  \put(4,1){\oval(3,1.8)[b]}
  \put(4,3){\oval(3,1.8)[r]}
  \put(3,1.5){\oval(1.8,0.4)}
\end{picture}
```



다음 명령은 알모양 곡선(oval)을 그린다.

```
\put(x,y){\oval(w,h)}
```

또는

```
\put(x,y){\oval(w,h)[position]}
```

(x,y) 는 곡선의 중심이다. 그리고 w 는 폭, h 는 높이다. 필요하다면 위치 인자 *position*를 지정할 수 있는데 여기에는 t(위쪽), b(바닥), l(왼쪽), r(오른쪽)이 올 수 있고 결합해서 쓸 수도 있다. 위의 예를 보라.

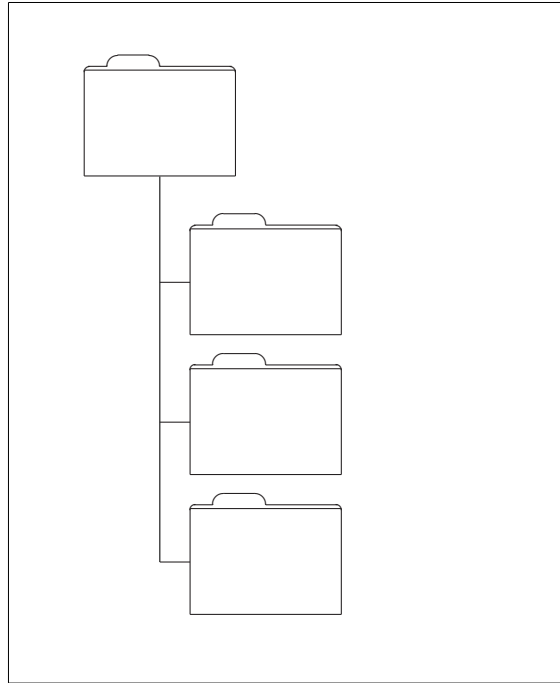
선 굵기를 제어하는 데는 두 종류의 명령이 있다. 한 가지는 `\linethickness{length}`이고, 또 한 가지는 `\thinlines`, `\thicklines`이다. `\linethickness{length}` 명령은 수평선과 수직선(그리고 2차 베지어 곡선)에만 효과가 나타난다. `\thinlines`와 `\thicklines`는 사선, 원, 알모양 곡선에 적용할 수 있다.

2.8 미리 정의한 그림 박스를 재사용하기

```

\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{\foldera}% declaration
\savebox{\foldera}
  (40,32)[bl]{% definition
    \multiput(0,0)(0,28){2}
      {\line(1,0){40}}
    \multiput(0,0)(40,0){2}
      {\line(0,1){28}}
    \put(1,28){\oval(2,2)[tl]}
    \put(1,29){\line(1,0){5}}
    \put(9,29){\oval(6,6)[tl]}
    \put(9,32){\line(1,0){8}}
    \put(17,29){\oval(6,6)[tr]}
    \put(20,29){\line(1,0){19}}
    \put(39,28){\oval(2,2)[tr]}
  }
\newsavebox{\folderb}% declaration
\savebox{\folderb}
  (40,32)[l]{% definition
    \put(0,14){\line(1,0){8}}
    \put(8,0){\usebox{\foldera}}
  }
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{\foldera}}
\multiput(34,86)(0,-37){3}
  {\usebox{\folderb}}
\end{picture}

```



다음 명령으로 그림 박스를 선언한다.

```
\newsavebox{name}
```

그 다음에 이 박스를 다음과 같이 정의한다.

```
\savebox{name}(width,height)[position]{content}
```

마지막으로 다음 명령으로 그리기를 한다.

```
\put(x,y)\usebox{name}
```

옵션 인자인 위치 인자 *position*은 미리 정의해 둔 박스의 “고정점”을 정의하는 효과가 있다. 위의 예에서 이것은 bl로 설정되어 있으므로 미리 정의해

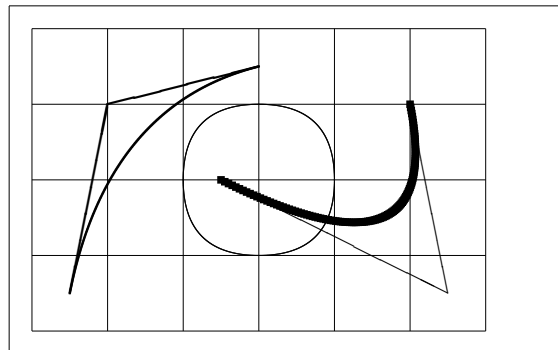
둔 박스의 고정점이 왼쪽 아래 귀퉁이가 될 것이다. 위치 설정인자로써는 이밖에도 `t`와 `r`이 더 있고 각각 ‘top’과 ‘right’를 의미한다.

`name` 인자는 \LaTeX 의 저장소 이름이다. 그러므로 명령어 형식을 띤다(위의 예에서 백슬래시로 시작하였다). 박스 그림 안에 또 다른 박스 그림을 넣을 수도 있다. 위의 예에서 `\foldera`는 `\folderb` 정의 안에서 쓰이고 있다.

`\line` 명령은 선분 길이가 약 3mm보다 작으면 동작하지 않기 때문에 `\oval` 명령을 써야 했다.

2.9 2차 베지어 곡선

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,4)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){7}
    {\line(0,1){4}}
  \multiput(0,0)(0,1){5}
    {\line(1,0){6}}
  \thicklines
  \put(0.5,0.5){\line(1,5){0.5}}
  \put(1,3){\line(4,1){2}}
  \qbezier(0.5,0.5)(1,3)(3,3.5)
  \thinlines
  \put(2.5,2){\line(2,-1){3}}
  \put(5.5,0.5){\line(-1,5){0.5}}
  \linethickness{1mm}
  \qbezier(2.5,2)(5.5,0.5)(5,3)
  \thinlines
  \qbezier(4,2)(4,3)(3,3)
  \qbezier(3,3)(2,3)(2,2)
  \qbezier(2,2)(2,1)(3,1)
  \qbezier(3,1)(4,1)(4,2)
\end{picture}
```



이 예에서 볼 수 있듯이 2차 베지어 곡선 네 개로 원을 만드는 것은 만족스럽지 못하며, 최소한 여덟 개가 필요하다. 이 그림은 또한 `\linethickness` 명령이 수평선과 수직선에만 적용되며 `\thinlines`와 `\thicklines`는 사선에 적용되는 것을 보여준다. 2차 베지어 곡선에는 두 종류의 명령이 다 적용된다는 것을 알 수 있다. 이 명령은 서로를 대체하는 효과를 가지므로 새로 선언이 이루어지면 이전의 것은 효력을 잃는다.

$P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ 가 끝점, m_1, m_2 가 2차 베지어 곡선의 각각의

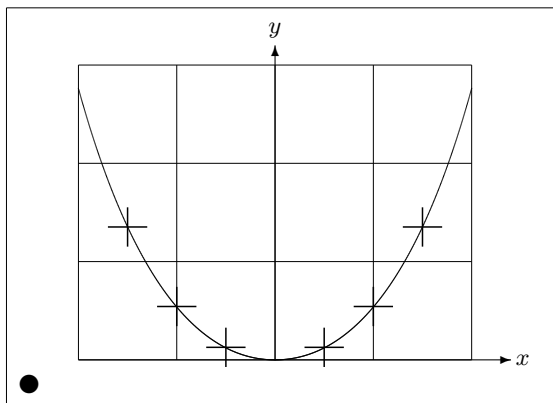
기울기를 나타낸다고 하자. 중간 조절점 $S = (x, y)$ 는 다음 식으로 구한다.

$$\begin{cases} x = \frac{m_2 x_2 - m_1 x_1 - (y_2 - y_1)}{m_2 - m_1}, \\ y = y_i + m_i(x - x_i) \quad (i = 1, 2). \end{cases} \quad (5.1)$$

필요한 `\qBezier` 명령행을 생성하는 Java 프로그램이 *Graphics in L^AT_EX 2_ε* [17]에 있다.

2.10 현수선

```
\setlength{\unitlength}{1.3cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
  \put(-2,0){\vector(1,0){4.4}}
  \put(2.45,-.05){\textit{x}}
  \put(0,0){\vector(0,1){3.2}}
  \put(0,3.35){\makebox(0,0){\textit{y}}}
  \qBezier(0.0,0.0)(1.2384,0.0)
    (2.0,2.7622)
  \qBezier(0.0,0.0)(-1.2384,0.0)
    (-2.0,2.7622)
  \linethickness{.075mm}
  \multiput(-2,0)(1,0){5}
    {\line(0,1){3}}
  \multiput(-2,0)(0,1){4}
    {\line(1,0){4}}
  \linethickness{.2mm}
  \put(.3,.12763){\line(1,0){.4}}
  \put(.5,-.07237){\line(0,1){.4}}
  \put(-.7,.12763){\line(1,0){.4}}
  \put(-.5,-.07237){\line(0,1){.4}}
  \put(.8,.54308){\line(1,0){.4}}
  \put(1,.34308){\line(0,1){.4}}
  \put(-1.2,.54308){\line(1,0){.4}}
  \put(-1,.34308){\line(0,1){.4}}
  \put(1.3,1.35241){\line(1,0){.4}}
  \put(1.5,1.15241){\line(0,1){.4}}
  \put(-1.7,1.35241){\line(1,0){.4}}
  \put(-1.5,1.15241){\line(0,1){.4}}
  \put(-2.5,-0.25){\circle*{0.2}}
\end{picture}
```



이 그림에서 현수선(catenary) $y = \cosh x - 1$ 에서 대칭되는 각 반쪽을 2차 베지어 곡선으로 근사시켰다. 곡선의 오른쪽 반은 점 $(2, 2.7622)$ 에서 끝나고 기울기는 $m = 3.6269$ 의 값을 갖는다. 식 (5.1)을 이용하여 중간 조절점을 계산할 수 있다. 그 결과는 $(1.2384, 0)$ 과 $(-1.2384, 0)$ 이다. 십자 표지는 실제

현수선 상의 점을 표시하는데 오차는 거의 식별하기 어려울 정도이고 1퍼센트 미만이다.

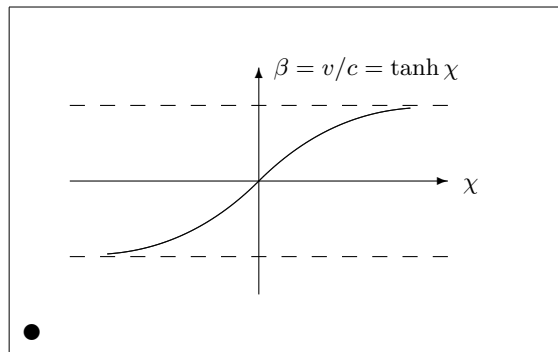
이 보기는 또한 `\begin{picture}` 명령의 선택 인자 사용법을 보여준다. `picture`는 임의의 “수학적” 좌표로 정의될 수 있다.

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

위의 명령은 검은 동그라미로 표시된 왼쪽 아래 모서리에 좌표 $(-2.5, -0.25)$ 를 부여한 것이다.

2.11 특수상대성이론에서 속도

```
\setlength{\unitlength}{1cm}
\begin{picture}(6,4)(-3,-2)
  \put(-2.5,0){\vector(1,0){5}}
  \put(2.7,-0.1){\mathchi}
  \put(0,-1.5){\vector(0,1){3}}
  \multiput(-2.5,1)(0.4,0){13}
    {\line(1,0){0.2}}
  \multiput(-2.5,-1)(0.4,0){13}
    {\line(1,0){0.2}}
  \put(0.2,1.4)
    {\mathbeta=v/c=\tanh\chi}
  \qbezier(0,0)(0.8853,0.8853)
    (2,0.9640)
  \qbezier(0,0)(-0.8853,-0.8853)
    (-2,-0.9640)
  \put(-3,-2){\circle*{0.2}}
\end{picture}
```



두 개의 베지어 곡선 조절점은 식 (5.1)로 계산한다. 양수 영역은 $P_1 = (0, 0)$, $m_1 = 1$ 과 $P_2 = (2, \tanh 2)$, $m_2 = 1/\cosh^2 2$ 로 결정된다. 여기서도 그림에 편의적으로 수학 좌표를 부여했는데 왼쪽 아래 모서리에 $(-3, -2)$ 를 부여했다(검은 동그라미로 표시하였다).

제 3 절 Xy-pic

By Alberto Manuel Brandão Simões <albie@alfarrabio.di.uminho.pt>

`xy`는 다이어그램을 그리기 위한 특별한 패키지이다. 간단히 문서의 전처리부에 다음과 같이 선언함으로써 사용할 수 있다.

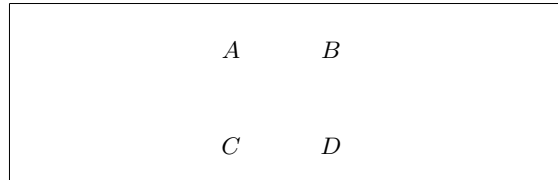
```
\usepackage[options]{xy}
```

`options`는 Xy-pic의 기능 중에서 사용하고자 하는 것을 열거하는 옵션이

다. 이 옵션은 패키지를 디버깅할 때 주로 유용하다. `all` 옵션을 지정할 것을 권장하는데 이렇게 하면 \LaTeX 은 모든 $\text{\texttt{Xy}}$ 명령을 불러들인다.

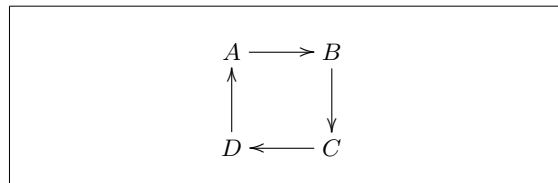
$\text{\texttt{Xy-pic}}$ 다이어그램은 행렬 형식의 캔버스에 그려진다. 각 다이어그램 요소들은 행렬 인자의 위치에 놓인다.

```
\begin{displaymath}
\mathrm{xymatrix}{A & B \\
& C & D }
\end{displaymath}
```



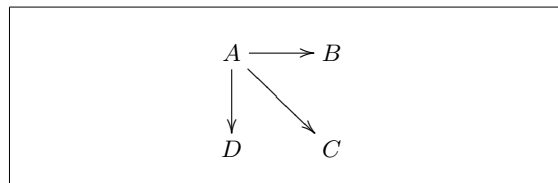
`\mathrm{xymatrix}` 명령은 수학 모드에서 사용하여야 한다. 여기서 2×2 행렬을 지시하였다. 이 행렬을 다이어그램으로 만들려면 `\ar` 명령으로 방향 화살표를 추가하면 된다.

```
\begin{displaymath}
\mathrm{xymatrix}{ A \ar[r] & B \ar[d] \\
D \ar[u] & C \ar[l] }
\end{displaymath}
```



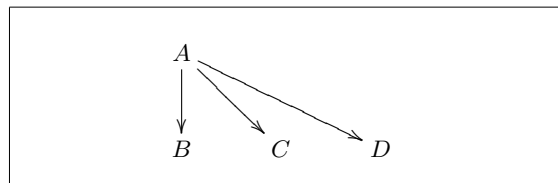
화살표 명령은 그 화살표가 시작하는 셀에 위치한다. 인자는 화살표의 방향을 표시하는데 (`up`, `down`, `right`, `left`)를 각각 나타낸다.

```
\begin{displaymath}
\mathrm{xymatrix}{
A \ar[d] \ar[dr] \ar[r] & B \\
D & C }
\end{displaymath}
```



대각선을 이어주려면 둘 이상의 방향을 결합한다. 큰 화살표를 그리기 위해서 같은 방향을 되풀이할 수도 있다.

```
\begin{displaymath}
\mathrm{xymatrix}{
A \ar[d] \ar[dr] \ar[dr] & & \\
B & & C & D }
\end{displaymath}
```

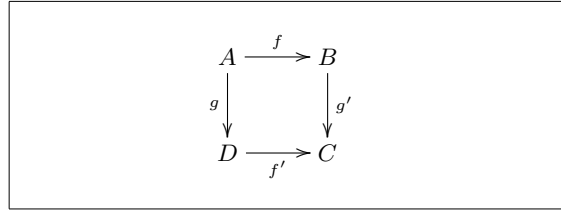


화살표에 라벨을 붙여서 훨씬 재미난 다이어그램을 그릴 수도 있다. 이를 위하여 일반적인 위첨자와 아래첨자 명령 기호를 사용한다.

```

\begin{displaymath}
\mathrm{xymatrix}
{
A \ar[r]^f \ar[d]_g & B \\
D \ar[r]_{f'} & C }
\end{displaymath}

```

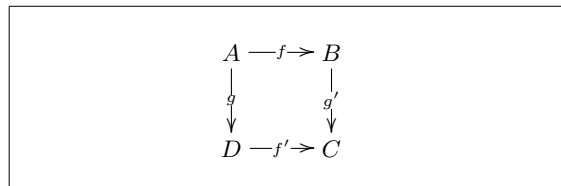


앞서 보인 대로 이 명령 기호들은 수학 모드에서 사용하는 것과 같다. 한 가지 차이점은 위첨자가 “화살표의 위”를, 아래첨자가 “화살표 아래”를 가리킨다는 점이다. 한 가지 명령 기호가 더 있는데 그것은 수직선 |이다. 이 명령 기호는 텍스트를 화살표 안에 놓는다.

```

\begin{displaymath}
\mathrm{xymatrix}
{
A \ar[r]|f \ar[d]|g & B \\
D \ar[r]|{f'} & C }
\end{displaymath}

```



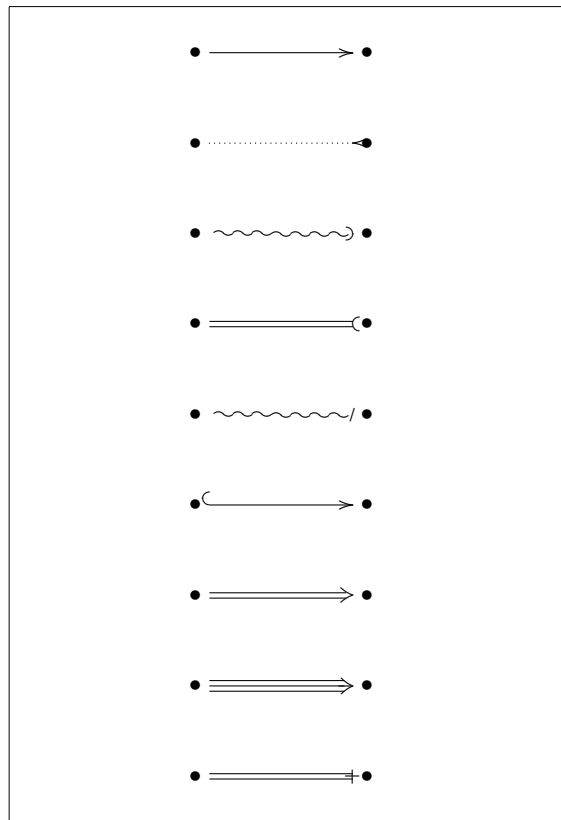
화살표에 동그라미를 치고 싶으면 `\ar[...]|hole`과 같이 한다.

어떤 경우에는 화살표의 유형을 구분하는 것이 중요할 때가 있다. 화살표 위에 라벨을 붙이거나 모양을 다르게 한다.

```

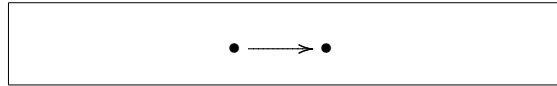
\shorthandoff{"}
\begin{displaymath}
\mathrm{xymatrix}
{
\bullet \ar@{->}[rr] & \bullet \\
\bullet \ar@{.>}[rr] & \bullet \\
\bullet \ar@{~}[rr] & \bullet \\
\bullet \ar@{=}[rr] & \bullet \\
\bullet \ar@{=}[rr] & \bullet \\
\bullet \ar@{=}[rr] & \bullet \\
\bullet \ar@{=}[rr] & \bullet \\
\bullet \ar@{=}[rr] & \bullet \\
\bullet \ar@{=}[rr] & \bullet \\
\bullet \ar@{=}[rr] & \bullet }
\end{displaymath}
\shorthandon{"}

```

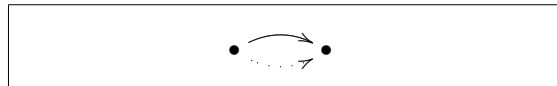


다음 두 다이어그램의 차이점에 주의하자.

```
\begin{displaymath}
\begin{matrix}
\bullet \longrightarrow \bullet \\
\bullet \longrightarrow \bullet
\end{matrix}
\end{displaymath}
```



```
\begin{displaymath}
\begin{matrix}
\bullet \curvearrowright \bullet \\
\bullet \curvearrowleft \bullet
\end{matrix}
\end{displaymath}
```



슬래시 사이에 오는 기호는 곡선이 그려지는 모양을 지정한다. `Xy-pic`은 곡선 모양을 바꾸는 여러 가지 방법을 제공하므로 더 자세한 사항은 `Xy-pic` 문서를 참고하기 바란다.

제 6 장

L^AT_EX을 자기에게 맞게 바꾸기

지금까지 배워온 명령들로 만들어진 문서는 대다수 독자에게 만족스럽게 보일 만할 것이다. 비록 아주 멋지지는 않더라도 좋은 활자 조판의 규칙을 따르고 있기 때문에 읽기 쉽고 보기 좋은 문서가 된다.

그러나, L^AT_EX에서 사용자가 원하는 결과를 얻을 수 있는 명령이나 환경을 제공하지 않는 경우도 있고, 기존의 명령을 써서 얻은 출력 결과가 사용자의 요구에 적합하지 않은 경우도 있다.

이 장에서는 L^AT_EX에 새로운 기능을 추가하려면 어떻게 하는지, 그리고 L^AT_EX 기본값을 이용했을 때 나오는 결과와 좀 다른 모양을 얻고 싶을 때 어떻게 하면 되는지에 대해 약간의 힌트를 제공하려 한다.

제 1 절 새로운 명령, 환경, 패키지

이 문서에서 소개하는 모든 명령어들은 모두 상자 안에 표시하고, 문서 마지막의 찾아보기(index)에 모아둔 것을 보았을 것이다. 이렇게 하기 위해 L^AT_EX 필수 명령들을 직접 사용하지 않고 새로운 명령과 환경을 정의하여 패키지를 만들었다. 이제 아래와 같이 간단하게 처리할 수 있다.

```
\begin{lscommand}  
\ci{dum}  
\end{lscommand}
```

\dum

이 예제에서 `lscommand`라는 이름의 새로운 환경을 사용하고 있는데, 이는 명령을 감싸는 상자를 그리는 구실을 한다. 그리고 `\ci`라는 이름의 새로운 명령은 명령어 이름을 조판하고 자동으로 찾아보기에 들어가도록 한다. 이 문서 마지막의 찾아보기를 보면 `\dum` 명령어가 나와 있고, `\dum` 명령을 언급한 모든 페이지를 가리키고 있는 것을 확인할 수 있을 것이다.

만약 명령을 상자 안에 넣는 방식이 마음에 들지 않는다면 `\scommand` 환경 정의를 살짝 바꾸어서 새로운 모양이 되도록 하면 된다. 그러면 이 환경을 사용한 모든 곳의 모양이 다 바뀌는데, 매번 박스를 그리고 명령을 써넣는 식으로 문서를 작성했을 때 그것을 일일이 다 찾아서 바꿔야 하는 것과 비교해보면, 이 방법이 얼마나 편리한 것인지 알 수 있다.

1.1 새로운 명령

자신의 새로운 명령을 추가하려면 다음과 같이 한다.

```
\newcommand{name}[num]{definition}
```

기본적으로 명령에는 두 개의 인자가 필요하다. 하나는 이 명령의 명칭 *name* 이고, 다른 하나는 이 명령의 정의 *definition*이다. 대괄호 안에 있는 *num* 인자는 옵션으로 여러 개의 인자를 가진 명령어를 만들 때 쓰는데, 최대 9개까지의 인자를 가지게 할 수 있다. 만약 이것을 쓰지 않으면 기본값은 0이다. 즉, 인자가 허용되지 않는다.

다음 두 예제는 새로운 명령을 만드는 데 도움을 줄 것이다. 첫 번째 예제에서는 `\tnss`라고 하는 새로운 명령을 만들었다. 이것은 "The Not So Short Introduction to L^AT_EX 2_ε"를 줄여 쓰게 한다. 만약 이 책의 제목을 문서 내에서 자주 사용한다면 이와 같이 줄여서 쓰면 편리하다.

```
\newcommand{\tnss}{The not  
so Short Introduction to  
LATEX}  
This is “\tnss” \ldots{  
“\tnss”
```

This is “The not so Short Introduction to L^AT_EX 2_ε” ... “The not so Short Introduction to L^AT_EX 2_ε”

다음 예제는 *num* 인자의 사용법을 나타낸다. #1은 사용자가 넣는 인자의 내용으로 치환되어 들어가게 된다. 만약 한 개 이상의 인자를 사용하려면, 두번째 인자부터 #2 ... 등으로 쓸 수 있다.

```
\newcommand{\txsit}[1]  
{This is the \emph{#1} Short  
Introduction to LATEX}  
% in the document body:  
\begin{itemize}  
\item \txsit{not so}  
\item \txsit{very}  
\end{itemize}
```

- This is the *not so* Short Introduction to L^AT_EX 2_ε
- This is the *very* Short Introduction to L^AT_EX 2_ε

새로운 명령을 만들었을 때 그 명령이 이미 정의되어 있는 것이라면, L^AT_EX은 이를 허용하지 않는다. 그러나 `\renewcommand` 명령을 사용하면 기존에 정

의된 내용은 무시되고 새로 정의한 명령이 유효하게 된다. `\renewcommand` 명령의 사용법은 `\newcommand`와 동일하다.

경우에 따라서 `\providecommand` 명령을 써야 할 때도 있다. 이것은 `\newcommand`와 유사하지만, 만약 명령이 이미 정의되어 있다면 \LaTeX 2_ϵ 은 새로 정의한 것을 무시하고 이미 정의되어 있는 것을 취한다.

\LaTeX 명령 뒤에 나오는 공백문자에 대해서 주의할 점이 있다. 더 자세한 내용은 6 쪽을 보라.

1.2 새로운 환경

`\newcommand` 명령과 마찬가지로 새로운 환경을 만드는 `\newenvironment` 명령이 있다. 이 명령의 사용법은 다음과 같다.

```
\newenvironment{name}[num]{before}{after}
```

`\newenvironment` 명령도 역시 옵션 인자를 가질 수 있다. *before* 인자에는 이 환경 안에 놓일 텍스트가 시작되기 전에 처리할 내용을 넣는다. *after* 인자는 `\end{name}` 명령을 만났을 때 실행될 내용이다.

아래 예제는 `\newenvironment` 명령의 사용법을 보여준다.

```
\newenvironment{king}
{\rule{1ex}{1ex}%
 \hspace{\stretch{1}}}
{\hspace{\stretch{1}}%
 \rule{1ex}{1ex}}
```

```
\begin{king}
My humble subjects \ldots
\end{king}
```

■ My humble subjects ... ■

num 인자의 사용법은 `\newcommand` 명령의 경우와 동일하다. \LaTeX 은 이미 정의되어 있는 환경이 또다시 정의되는 것을 허용하지 않는데, 만약 기존의 환경을 재정의하고자 한다면, `\renewenvironment` 명령을 사용하면 된다. 이 명령의 문법은 `\newenvironment` 명령과 동일하다.

위의 예제에서 사용된 명령들은 나중에 설명하기로 한다. `\rule` 명령은 124 쪽에서 설명하며, `\stretch`는 117 쪽에서 설명한다. `\hspace`에 관한 더 자세한 내용은 117 쪽에서 볼 수 있다.

1.3 여분의 공백

새로운 환경을 만들 때 불필요한 공백이 뜻하지 않게 들어가서 치명적인 결과를 가져오는 수가 있다. 예를 들어보자면, 표제 환경을 만들고 그 표제는

들여쓰기를 하지 않게 하면서 그 다음에 이어지는 단락도 들여쓰기 되지 않도록 하려 한다고 하자. 환경의 시작 블록(block)에 `\ignorespaces` 명령을 두어서 환경이 시작된 후의 공백을 무시하도록 하여야 한다. 종료 블록은 신경써야 할 것이 좀 더 있는데 환경이 종료될 때는 약간의 특별한 처리과정이 일어나기 때문이다. `\ignorespacesafterend` 명령을 주면 L^AT_EX은 ‘환경 종료 처리과정’을 마친 다음에 `\ignorespaces` 명령을 실행하게 된다. 다음 예를 통해 무슨 일이 일어나는 것인지 알아보자.

```
\newenvironment{simple}%
{\noindent}%
{\par\noindent}

\begin{simple}
See the space\\to the left.
\end{simple}
Same\\here.
```

See the space
to the left.

Same
here.

```
\newenvironment{correct}%
{\noindent\ignorespaces}%
{\par\noindent\ignorespacesafterend}

\begin{correct}
No space\\to the left.
\end{correct}
Same\\here.
```

No space
to the left.

Same
here.

1.4 L^AT_EX 명령행 조작

Unix와 같은 운영체제에서 작업하는 경우에는 여러 개의 파일로 이루어진 큰 규모의 문서를 일괄 조성하기 위해 `Makefile`을 사용할 수 있다. 이와 관련하여 L^AT_EX을 명령행에서 서로 다른 매개변수 값을 주어 호출함으로써 동일한 문서의 여러 판본을 한꺼번에 만들어내는 재미있는 일이 가능하다. 다음 문장을 문서에 추가해보자.

```
\usepackage{ifthen}
\ifthenelse{\equal{\blackandwhite}{true}}{
  % "black and white" mode; do something..
}{
  % "color" mode; do something different..
}
```

이제 다음과 같이 L^AT_EX을 호출한다.

```
latex '\newcommand{\blackandwhite}{true}\input{test.tex}'
```

우선 `\blackandwhite` 명령을 정의하고 그 후에 실제 파일을 `\input` 명령으로 읽어들인다. `\blackandwhite` 명령을 `false`로 설정함으로써 문서를 흑백이 아닌 칼라 버전으로 만들 수 있다. 물론 `\blackandwhite` 값에 따라 흑백과 칼라 버전 문서를 어떤 식으로 만들 것인지는 미리 문서에 코딩해두어야 할 것이다.

1.5 사용자 패키지

만약 새로운 환경과 명령을 많이 정의한다면, 문서의 전처리부가 무척 길어질 수 있다. 이런 경우 환경과 명령의 정의를 담고 있는 \LaTeX 패키지를 만들어 쓰면 좋다. 패키지로 만든 다음에는 `\usepackage` 명령으로 이 패키지를 문서에 포함하여 사용할 수 있다.

```
% Demo Package by Tobias Oetiker
\ProvidesPackage{demopack}
\newcommand{\tnss}{The not so Short Introduction to \LaTeXe}
\newcommand{\txsit}[1]{The \emph{#1} Short
                      Introduction to \LaTeXe}
\newenvironment{king}{\begin{quote}}{\end{quote}}
```

그림 6.1: Example Package.

패키지를 만드는 것은 기본적으로 문서의 전처리부에 있는 내용을 독립 파일로 복사하는 것이다. 이 파일의 확장명은 `.sty`가 된다. 다만 한 가지 특별한 명령이 있는데,

`\ProvidesPackage{package name}`

이 명령은 패키지 파일의 가장 앞 부분에서 사용한다. `\ProvidesPackage`는 패키지의 이름을 \LaTeX 에 알리는 기능을 함으로써, 컴파일 시 같은 스타일을 두 번 이상 삽입하려 할 때 에러 메시지로 이를 알려주기 위해 사용한다. 그림 6.1은 위의 예제에서 정의된 명령들을 포함하는 간단한 예제 패키지를 보여준다.

제 2 절 글꼴과 크기

2.1 글꼴 바꾸기 명령

\LaTeX 은 문서의 논리적 구조(장/절, 각주, ...)에 따라 적절한 글꼴 및 글꼴크기를 선택한다. 경우에 따라 사용자가 임의로 글꼴의 모양과 크기를 바꾸고

싶을 경우가 있다. 이에 관한 명령들이 표 6.1과 6.2에 있다. 각 글꼴의 실제 크기는 설계하기 나름이며, 이는 문서 클래스와 클래스 옵션에 따라 달라진다. 각 표준 문서 클래스에서 이용하는 글꼴 선택 명령의 절대 포인트 크기를 표 6.3에 요약해 두었다.

```
{\small The small and
\textbf{bold} Romans ruled}
{\Large all of great big
\textit{Italy}.}
```

The small and **bold** Romans ruled all of great big *Italy*.

L^AT_EX 2_ε의 한가지 중요한 특징은 글꼴의 속성이 비의존적이라는 것이다. 달리 말하자면, 이전에 설정한 볼드(굵게)/슬랜트(기울임) 속성을 유지하면서 크기는 물론 글꼴 자체도 바꿀 수 있다는 뜻이다.

수식 모드에서 글꼴 바꾸기 명령을 사용하여 수식 모드를 일시적으로 빠져 나가 일반 텍스트로 들어갈 수도 있다. 만일 수식 조판용 글꼴을 바꾸고자 한다면 수식 글꼴 바꾸기를 위하여 특별히 마련된 명령을 써야 한다. 이것은 표 6.4를 참조하라.

글꼴 크기 명령에서 중괄호({})는 중요한 역할을 한다. 이렇게 중괄호로 묶으면 그 안의 내용이 하나의 그룹을 이루게 된다. 대부분의 L^AT_EX 명령은 그룹

표 6.1: 글꼴

<code>\textrm{...}</code>	roman	<code>\textsf{...}</code>	sans serif
<code>\texttt{...}</code>	typewriter		
<code>\textmd{...}</code>	medium	<code>\textbf{...}</code>	bold face
<code>\textup{...}</code>	upright	<code>\textit{...}</code>	<i>italic</i>
<code>\textsl{...}</code>	<i>slanted</i>	<code>\textsc{...}</code>	SMALL CAPS
<code>\emph{...}</code>	<i>emphasized</i>	<code>\textnormal{...}</code>	document font

표 6.2: 글꼴 크기

<code>\tiny</code>	tiny font	<code>\Large</code>	larger font
<code>\scriptsize</code>	very small font	<code>\LARGE</code>	very large font
<code>\footnotesize</code>	quite small font		
<code>\small</code>	small font	<code>\huge</code>	huge
<code>\normalsize</code>	normal font	<code>\Huge</code>	largest
<code>\large</code>	large font		

표 6.3: 클래스별 글꼴 절대 크기

size	10pt (default)	11pt option	12pt option
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

안에서 쓰이면 효력이 그 범위 안으로 제한된다.

He likes `{\LARGE large and
{\small small} letters}`.

He likes large and small letters.

글꼴 크기 명령은 줄 간격도 바꾼다. 다만 글꼴 크기 명령의 효력 범위 안에 문단 끝(`\par` 또는 `\\`)이 올 때만 줄 간격이 바뀐다. 따라서 범위의 끝을 나타내는 중괄호 `}`가 너무 일찍 나타나면 줄 간격이 바뀌지 않는다. 다음 두 예에서 `\par` 명령¹의 위치를 주의깊게 살펴보자.

¹`\par` 명령은 빈 줄 하나를 넣는 것과 같다.

표 6.4: 수학 글꼴

<code>\mathrm{...}</code>	Roman Font
<code>\mathbf{...}</code>	Boldface Font
<code>\mathsf{...}</code>	Sans Serif Font
<code>\mathtt{...}</code>	Typewriter Font
<code>\mathit{...}</code>	<i>Italic Font</i>
<code>\mathcal{...}</code>	<i>CALLIGRAPHIC FONT</i>
<code>\mathnormal{...}</code>	<i>Normal Font</i>

```
{\Large Don't read this! It is not
true. You can believe me!}\par}
```

Don't read this! It is not true.
You can believe me!

```
{\Large This is not true either.
But remember I am a liar.}\par}
```

This is not true either. But re-
member I am a liar.

한 문단 전체나 그보다 더 긴 텍스트의 일정 부분에 대해 글꼴 크기를 바꾸고자 한다면, 환경(environment) 형태의 구문법을 사용하여 글꼴을 바꾸는 것이 나을 수도 있다.

```
\begin{Large}
This is not true.
But then again, what is these
days \ldots
\end{Large}
```

This is not true. But then
again, what is these days ...

이렇게 하면 범위를 정해주기 위한 중괄호를 덜 사용해도 된다.

2.2 경고! 경고!

이 장을 시작할 때 말한 대로, 문서의 각 부분에 이와 같이 구체적이고 개별적인 명령을 어지럽게 삽입하는 것은 위험스런 일이다. 이런 방식은 L^AT_EX의 기본 개념과 상충하기 때문이다. L^AT_EX은 문서의 논리적 구성과 그 결과물의 실제 모양을 구별해서 취급한다는 개념을 가지고 있다. 다시 말하면 똑같은 내용의 글꼴 바꾸기 명령이 여러 번 나오는 경우라면 그 명령을 매번 써넣을 것이 아니라 논리적으로 글꼴 바꾸기를 의미하는 `\newcommand`를 정의해서 사용해야 한다는 뜻이다.

```
\newcommand{\oops}[1]{\textbf{#1}}
Do not \oops{enter} this room,
it's occupied by a \oops{machine}
of unknown origin and purpose.
```

Do not **enter** this room, it's occupied by a
machine of unknown origin and purpose.

이러한 접근은 훗날 위험(물)을 나타내기 위해 `\textbf`를 쓰지 않고 좀 더 시각적 표현을 하고자 할 때 유용하다. `\textbf`를 하나하나 써넣어서 만든 문서라면 그것을 매번 찾아서 확인하고 바꾸어야 할 것이다. 게다가 `\textbf`로 표시된 모든 단어들이 꼭 위험(물)을 나타내는 단어일 것이라는 법이 없으므로 그것들을 일일이 구별해서 바꾸어야 한다. 이에 비하면 `\oops`를 정의해서 한번에 위험물에 대한 표현을 바꾸는 것이 훨씬 쉽고 편리하다.

2.3 조언

글꼴과 글자크기에 대한 이야기를 끝내기 전에 조언 한 마디.²

Remember! *The MORE fonts you use in a document, the more READABLE and beautiful it becomes.*

제 3 절 간격

3.1 행간

만약 문서의 행간을 더 벌리고 싶다면, 다음과 같이 `linespread` 값을 변경한다.

```
\linespread{factor}
```

이 명령은 문서의 전처리부에 둔다. “한 줄 반” 행간을 적용하려면 *factor* 인자 값을 1.3으로 하고, “두 줄” 행간으로 하려면(원고를 배행간으로 요구하는 경우도 많다) `\linespread{1.6}`으로 한다. 영문 문서에서는 행간을 넓게 벌리지 않는 것이 일반적이므로 행간 기본값은 1이다.³

`\linespread` 명령의 효과는 좀 지나친 감이 있어서 문서 전체에 미친다. 일부 문단만 행간을 바꾸고자 할 수도 있을 것이다. 이럴 경우 `\baselineskip` 값을 일시적으로 조정하는 방법을 쓰는 사람이 있다.

```
\setlength{\baselineskip}{1.5\baselineskip}
```

```
{\setlength{\baselineskip}%
{1.5\baselineskip}
This paragraph is typeset with the
baseline skip set to 1.5 of what
it was before. Note the par command
at the end of the paragraph.\par}
```

This paragraph has a clear purpose,
it shows that after the curly brace
has been closed, everything is back
to normal.

This paragraph is typeset with the baseline skip set to 1.5 of what it was before. Note the par command at the end of the paragraph.

This paragraph has a clear purpose, it shows that after the curly brace has been closed, everything is back to normal.

일부 문단에 대한 행간 조절은 `setspace` 패키지를 이용하는 편이 낫다.

²적어도 한글 문서에 관한 한, 지나치게 많은 글꼴을 하나의 문서에서 사용하는 것은 자칫 문서의 품위를 떨어뜨리고 내용에 집중할 수 없도록 만들 위험이 있다. —[역주]

³한글 문서에서는 배행간에 적절한 값이 1.75라고 알려져 있다. 그리고 영문 문서와 달리 보통 1.3 정도의 값을 주는 것이 표준적인 행간으로 받아들여진다. —[역주]

3.2 문단 모양

L^AT_EX에서 문단 모양에 영향을 주는 파라미터가 두 개 있다. 다음과 같이 그 값을 설정할 수 있다.

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

입력 파일의 전처리부에 위와 같은 설정을 둬으로써 문단의 레이아웃을 바꿀 수 있다. 위의 보기가 의미하는 것은 문단의 첫 줄 들여쓰기 값을 0으로 맞추면서 문단 사이의 수직 공백을 늘려주는 것이다.

위 예제에서 `plus`와 `minus` 길이값이 가리키는 것은 T_EX이 페이지를 조판하는 과정에서 페이지 세로길이에 정확하게 맞출 수 없을 때 문단 사이의 간격을 줄이거나 늘릴 수 있는 허용범위를 지정해주는 것이다.

유럽에서는 일반적으로 문단과 문단 사이를 넓히고 들여쓰기를 하지 않는다. 이 두 명령은 차례에도 영향을 미치기 때문에 차례의 줄 사이가 너무 벌어지는 경우가 있다. 이것을 피하려면 이 두 명령을 전처리부에 놓지 않고 문서 본문의 `\tableofcontents` 다음에 넣는 방법이 있다. 아니면 아예 이 설정을 사용하지 않는 것도 한 가지 방법이다. 전문적으로 조판이 잘 된 책을 보면 첫 줄 들여쓰기를 하고 문단과 문단 사이는 그대로 두는 것이 일반적이다.

첫 줄 들여쓰기되지 않은 문단을 강제로 들여쓰기하려면 다음 명령을 문단 시작 부분에 넣는다.⁴

```
\indent
```

말할 것도 없이 `\parindent`의 값은 0 이 아닌 값으로 설정되어야 제대로 동작한다.

첫 줄 들여쓰기를 하지 않으려면 다음 명령을 문단의 시작 부분에 넣는다.

```
\noindent
```

`\section` 명령 없이 바로 본문을 시작하고 싶을 때 이 명령을 쓰면 편리하다.

⁴각 장(절)의 첫 문단을 들여쓰기 하려면 `indentfirst` 패키지를 사용한다. 이 패키지는 ‘tools’ 패키지 묶음에 포함되어 있다.

3.3 수평 간격

L^AT_EX은 단어간 간격과 문장 사이 간격을 자동으로 설정한다. 임의의 수평 간격을 강제로 주려면 다음과 같이 한다.

```
\hspace{length}
```

`\hspace` 대신 (별표 붙은) `\hspace*` 명령을 사용하면 공백을 넣어야 하는 곳이 줄 끝이나 줄 처음이라도 간격이 유지된다. *length* 인자는 보통 숫자 하나에 단위가 붙은 꼴인데, 주요 단위를 표 6.5에 나타내었다.

```
This\hspace{1.5cm}is a space  
of 1.5 cm.
```

```
This          is a space of 1.5 cm.
```

줄의 끝까지 수평 간격을 넣어 채우는 가변 길이 명령이 있다.

```
\stretch{n}
```

두 개의 `\hspace{\stretch{n}}` 명령이 같은 줄에서 사용되면 stretch 인수 *n* 값에 비례해서 간격이 늘어나게 된다.

```
x\hspace{\stretch{1}}  
x\hspace{\stretch{3}}x
```

```
x          x          x
```

수평 간격을 텍스트와 함께 사용하는 경우 현재의 기본 글꼴 크기에 비례 하도록 간격을 설정하는 것이 좋다. 이는 가변 길이 단위 `em`과 `ex`를 사용하면 가능하다.

```
{\Large{}big\hspace{1em}y}\  
{\tiny{}tin\hspace{1em}y}
```

```
big  y  
tin  y
```

3.4 수직 간격

문단(paragraph), 절(section), 소절(subsection) 등의 수직 간격은 L^AT_EX에 의해 자동으로 설정된다. 두 문단 사이에 임의의 수직 간격을 두어야 할 필요가 있으면 다음 명령을 쓴다.

```
\vspace{length}
```

이 명령의 전후에 각각 빈 줄을 하나씩 두는 것이 보통이다. 만약 페이지의 첫 줄 또는 마지막 줄에 걸치더라도 주어진 간격만큼 강제로 벌려야 한다면 `\vspace` 대신에 별표 붙은 `\vspace*` 명령을 사용한다.

표 6.5: T_EX의 길이단위

mm	millimetre $\approx 1/25$ inch	□
cm	centimetre = 10 mm	□
in	inch = 25.4 mm	□
pt	point $\approx 1/72$ inch $\approx \frac{1}{3}$ mm	□
em	approx width of an ‘M’ in the current font	□
ex	approx height of an ‘x’ in the current font	□

`\stretch` 명령을 `\pagebreak`와 함께 쓰면 텍스트를 페이지의 마지막 줄에 놓거나 한 페이지의 중앙에 둘 수 있다.

```
Some text \ldots
```

```
\vspace{\stretch{1}}
```

```
This goes onto the last line of the page.\pagebreak
```

새로운 문단을 시작하지 않고 한 문단 안에서 두 줄 사이를 띄고 싶거나 표에서 두 줄 사이의 간격을 넓히려면 다음 명령

`\[length]`

를 사용한다.

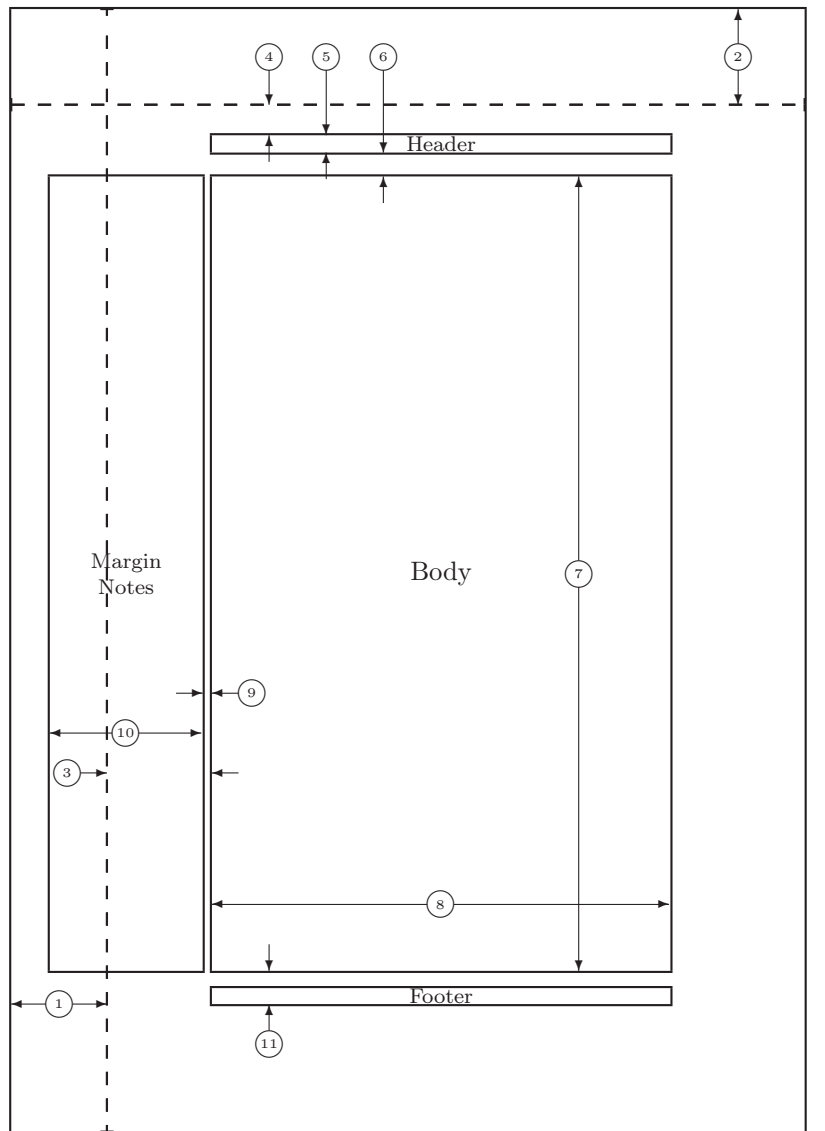
`\bigskip`과 `\smallskip` 명령은 정확한 값(수치)을 특정하지 않아도 미리 정의된 크기만큼 수직 간격을 띄울 수 있다.

제 4 절 페이지 레이아웃

L^AT_EX 2_ε에서는 `\documentclass` 명령의 인자로 용지 크기를 지정할 수 있다. 그러면 텍스트 여백을 적절하게 자동으로 설정하지만, 이렇게 설정된 값이 마음에 들지 않는 때도 있을 것이다. 당연히 이 값을 변경하는 것도 가능하다. 그림 6.2는 변경할 수 있는 파라미터를 모두 보여준다. 이 그림은 ‘tools’ 패키지 묶음의 layout 패키지를 써서 만든 것이다.⁵

잠깐! ... 이 레이아웃이 너무 좁아 보여서 좀 넓게 만들어야겠다는 분에게 한 마디 하겠다. 잠시 생각해 보자. 다른 L^AT_EX 기능들이 대부분 그렇듯이, 이처럼 페이지를 좁게 짜는 것이 기본값으로 되어 있는 데는 그럴 만한 이유가 있는 것이다.

⁵macros/latex/required/tools



1	one inch + \hoffset	2	one inch + \voffset
3	\oddsidemargin = 28pt or \evensidemargin	4	\topmargin = 23pt
5	\headheight = 13pt	6	\headsep = 18pt
7	\textheight = 598pt	8	\textwidth = 345pt
9	\marginparsep = 7pt	10	\marginparwidth = 115pt
11	\footskip = 25pt		\marginparpush = 5pt (not shown)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

그림 6.2: 페이지 레이아웃 파라미터

MS 워드의 기본 규격 레이아웃과 비교해보면, 확실히 L^AT_EX의 기본 레이아웃이 상당히 좁다. 그러나 책⁶을 펴서 표준 텍스트 행 한 줄에 몇 글자 정도가 들어있는지 세어 보라. 한 줄에 많아야 66자(영문)를 넘지 않을 것이다. 이제 L^AT_EX 문서 한 줄에 몇 자나 들어가는지 세어 보라. 역시 약 66자 정도가 들어가는 것을 알 수 있을 것이다. 실험을 해보면 한 줄에 들어가는 문자가 많을수록 읽기가 어려워진다는 걸 알 수 있다. 글자가 많을수록 사람의 눈이 줄 끝에서 다음 줄 첫 글자로 이동하기가 어려워지기 때문이다. 신문이 여러 단으로 조판되는 것도 이러한 이유 때문이다.

그러므로 만약 행의 폭을 늘리면 독자들이 괴로워진다는 것을 명심하라. 그래도 굳이 문단폭을 넓혀야겠다면, 어떻게 하면 되는지 가르쳐주겠다...

L^AT_EX은 페이지 레이아웃 파라미터 값을 바꾸는 두 개의 명령어를 제공한다. 이 명령들은 문서의 전처리부에서 사용한다.

첫번째는 어떤 고정값을 특정 변수에 할당하는 것이다.

```
\setlength{parameter}{length}
```

두번째 것은 어떤 변수의 기존값에 일정한 값을 더하는 것이다.

```
\addtolength{parameter}{length}
```

두번째 명령은 기존값에 대하여 상대 설정이 가능하므로 `\setlength` 명령어보다 더 유용하다. 예를 들어, 텍스트 영역의 문단폭을 1 센티미터 더 늘리려면 전처리부에 다음과 같이 쓰면 된다.

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

이런 일을 하는 데 `calc` 패키지가 도움이 될 수 있다. 이것은 `\setlength`의 인수에 대해 산술연산을 할 수 있게 해준다. 길이 설정뿐 아니라 그밖의 다른 함수의 경우에도 인자로서 숫자를 쓸 때는 이 패키지의 도움을 받을 수 있다.

제 5 절 길이 문제, 몇 가지 더

L^AT_EX 문서에서는 되도록 절대 길이를 사용하지 않는 것이 좋다. 길이를 지정할 때는 쪽 레이아웃을 구성하는 다른 요소(예를 들면 문단 폭과 같은)의 값을 기초로 해서 상대값을 지정하는 것이 낫다. 그림을 페이지 너비에 맞게 채우고 싶으면 `\textwidth`를 그림의 폭으로 쓰면 될 것이다.

⁶저명한 출판사에서 나온 진짜 인쇄본의 책.

다음 세 개의 명령으로 어떤 문자열(*text*)의 폭 · 높이 · 깊이를 변수 *variable*에 저장할 수 있다.

```
\settoheight{variable}{text}
\settodepth{variable}{text}
\settowidth{variable}{text}
```

다음 예제는 이 명령을 어떻게 활용할 수 있는지 보여준다.

```
\flushleft
\newenvironment{vardesc}[1]{%
  \settowidth{\parindent}{#1:\ }
  \makebox[0pt][r]{#1:\ }}{}

\begin{displaymath}
a^2+b^2=c^2
\end{displaymath}

\begin{vardesc}{Where}$a$,
$b$ -- are adjoin to the right
angle of a right-angled triangle.

$c$ -- is the hypotenuse of
the triangle and feels lonely.

$d$ -- finally does not show up
here at all. Isn't that puzzling?
\end{vardesc}
```

$$a^2 + b^2 = c^2$$

Where: *a*, *b* – are adjoin to the right angle of a right-angled triangle.

c – is the hypotenuse of the triangle and feels lonely.

d – finally does not show up here at all. Isn't that puzzling?

위의 코드에서 `vardesc`라는 환경이 새로 정의되었다. `\settowidth` 명령으로 `\parindent` 길이 변수에 인자(#1)로 주어진 문자열의 길이와 `:\`에 해당하는 길이가 저장되었다. 그러므로 이후의 문단은 이 길이값만큼 들여쓰기될 것이다. 실제 환경의 첫번째 인자로 `Where`가 주어졌기 때문에 이 단어 전체와 그 뒤의 간격에 해당하는 길이가 그 다음 단락부터 들여쓰기 값으로 사용되고 있음을 볼 수 있을 것이다. `\settowidth` 등에 사용할 수 있는 변수는 길이값이기만 하면 되기 때문에 사용자가 `\newlength`로 새로이 정의한 변수를 사용할 수 있다.

제 6 절 박스

L^AT_EX은 상자를 하나씩 쌓아올려서 페이지를 만들어간다. 우선 글자 하나하나가 작은 상자이고, 단어는 문자 상자들을 잘 붙여서 만든 상자이다. 이 단어 상자를 다시 다른 단어 상자과 붙여간다. 단어와 단어를 붙일 때는 약간 특별한 방법을 쓰는데, 이어지는 단어들이 쪽과 줄에 딱 맞아들어가도록 신축성

있게 그 간격을 늘리거나 줄이면서 붙여가는 것이다.

실제로 이루어지는 일은 이보다 더 복잡하지만, 요는 T_EX이 상자를 붙여가는 방법으로 식자한다는 것이다. 글자만이 상자인 것은 아니다. 상자에는 무엇이든지 들어갈 수 있다. 심지어 상자 그 자체도 상자에 담을 수 있다. 이렇게 채워진 상자를 L^AT_EX은 각각 한 글자인 것처럼 취급한다.

상자라고 말하진 않았어도, 앞 장에서 이미 상자를 다루어 본 적이 있다. 예를 들면, `tabular` 환경이나 `\includegraphics` 명령은 상자를 생성한다. 즉 두 개의 표나 그림을 나란히 붙도록 배열하는 것도 가능하다는 뜻이다. 단지 이렇게 할 때 두 개의 그림이나 표의 폭을 합한 값이 본문폭(`textwidth`)보다 더 넓어서는 안 된다.

한 문단을 상자 안에 넣으려면 다음 두 가지 방법이 있다. 명령 구문을 이용할 때는 다음과 같이 하고,

```
\parbox[pos]{width}{text}
```

‘환경’ 구문을 쓰고 싶으면 다음과 같이 한다.

```
\begin{minipage}[pos]{width} text \end{minipage}
```

`pos` 인자로 올 수 있는 것은 상자의 수직 정렬을 조절하는 `c`, `t`나 `b` 문자 중 하나이다. `width` 인자는 상자의 폭을 지정하는 것이다. `minipage`와 `parbox`의 가장 중요한 차이점은, `parbox` 안에서는 사용하지 못하는 명령과 환경이 좀 있지만 `minipage` 안에서는 거의 모든 명령이나 환경을 사용할 수 있다는 점이다.

`\parbox`는 줄바꿈 등등이 포함된 문단 전체를 상자에 넣는다. 수평으로 배열된 것들만을 하나로 묶는 상자 만들기 명령도 있다. 앞에서 본 바 있는 `\mbox`라는 것이 그 가운데 하나인데, 이 명령은 여러 개의 상자를 한 개의 상자로 묶어주기만 할 뿐이다. 그래서 `\mbox`로 묶인 단어 사이에서는 줄바꿈이 일어나지 않는다. 상자를 또다른 상자에 넣을 수 있기 때문에, 다음과 같은 수평 상자 묶음 명령을 잘 활용하면 매우 유연하게 써먹을 수 있다.

```
\makebox[width][pos]{text}
```

`width` 값은 밖에서 보았을 때 그려지는 상자의 폭이다.⁷ 직접 길이값을 써넣지 않고 `\width`, `\height`, `\depth`, `\totalheight`라고 지정할 수도 있다. 이렇게 하면 이 값들은 `text`를 식자한 결과 얻어지는 값을 계산하여 취한다. `pos` 인자로써 `c`, `l`, `r`, `s` 문자 가운데 하나를 취할 수 있는데, 각각 `center`(중앙

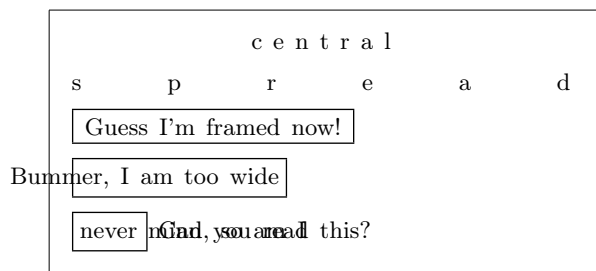
⁷즉, 이 값이 상자 안의 내용보다 작은 값일 수도 있다. 이 폭을 0pt로 설정할 수도 있는데, 이 때는 박스 안의 텍스트는 바깥쪽 상자와는 관계없이 그냥 출력될 것이다.

으로), `flushleft` (왼쪽으로 몰기), `flushright` (오른쪽으로 몰기), 또는 `spread` (크기에 맞추어 펼치기)를 뜻한다.

`\framebox` 명령의 기능은 `\makebox`와 동일하지만, 텍스트 둘레에 상자를 그려준다.

다음 예제는 `\makebox`와 `\framebox` 명령을 활용하여 어떤 일을 할 수 있는지 보여주는 것이다.

```
\makebox[\textwidth]{%
  c e n t r a l}\par
\makebox[\textwidth][s]{%
  s p r e a d}\par
\framebox[1.1\width]{Guess I'm
  framed now!} \par
\framebox[0.8\width][r]{Bummer,
  I am too wide} \par
\framebox[1cm][l]{never
  mind, so am I}
Can you read this?
```

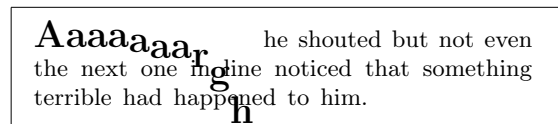


여기까지 수평(horizontal) 위치를 조절하는 방법을 보았다. 다음으로 수직(vertical) 위치를 조절하는 방법을 알아보자.⁸ 이것도 \LaTeX 에서는 어렵지 않다.

```
\raisebox{lift}[extend-above-baseline][extend-below-baseline]{text}
```

이 명령은 상자의 수직 속성을 정의하게 한다. 첫 세 개의 인자에는 고정값을 넣어도 되지만, `\width`, `\height`, `\depth`, `\totalheight`를 써서 `text`의 크기를 계산하여 그 값으로 동작하게 할 수도 있다.

```
\raisebox{0pt}[0pt][0pt]{\Large%
\textbf{Aaaa}\raisebox{-0.3ex}{a}%
\raisebox{-0.7ex}{aa}%
\raisebox{-1.2ex}{r}%
\raisebox{-2.2ex}{g}%
\raisebox{-4.5ex}{h}}
he shouted but not even the next
one in line noticed that something
terrible had happened to him.
```



⁸수직, 수평 위치를 모두 다루려면 이 각각을 조절하면 된다.

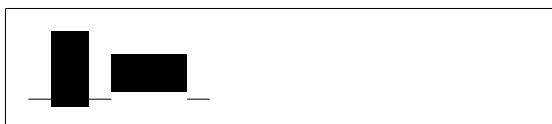
제 7절 선 그리기

이 앞 몇 페이지 전에 아래와 같은 명령을 만난 적이 있을 것이다.

```
\rule[lift]{width}{height}
```

이명령은 보통 간단한 검정 상자를 그리는 데 사용한다.

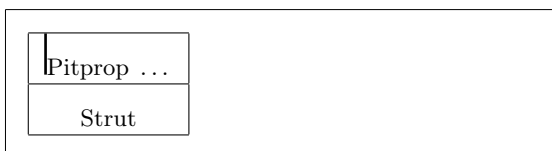
```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



이것은 수직선이나 수평선을 그릴때 유용하다. 예를 들면 이 책 겉표지의 선은 `\rule` 명령을 사용해서 만든 것이다.

높이는 있으나 폭이 없는 특별한 선(`\rule{0pt}{height}`)이 쓰일 때가 있다. 전문 조판 용어로 `strut`라는 것이다. 이것은 페이지 상의 어떤 요소가 최소한 일정한 높이를 갖게 하기 위해서 쓰인다. `tabular` 환경에서 행(row) 높이가 적어도 일정값 이상을 갖도록 하는 데 쓸 수 있다.

```
\begin{tabular}{|c|}
\hline
\rule{1pt}{4ex}Pitprop \ldots\\
\hline
\rule{0pt}{4ex}Strut\\
\hline
\end{tabular}
```



끝

참고 문헌

- [1] Leslie Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994, ISBN 0-201-52983-1.
- [2] Donald E. Knuth. *The TeXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley, Reading, Massachusetts, second edition, 1984, ISBN 0-201-13448-9.
- [3] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LaTeX Companion, (2nd Edition)*. Addison-Wesley, Reading, Massachusetts, 2004, ISBN 0-201-36299-6.
- [4] Michel Goossens, Sebastian Rahtz and Frank Mittelbach. *The LaTeX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.
- [5] LaTeX을 설치하면 이른바 *LaTeX Local Guide*라는 문서를 제공하게 되어 있다. 여기에는 로컬 시스템에 특정한 사항을 설명해둔다. 이 문서는 `local.tex`이라는 문서로 작성되어야 한다. 불행히도 게으른 시스템 관리자가 이 문서를 작성하지 않는 경우가 있다. 그럴 때는 주변의 LaTeX 도사를 찾아서 도움을 구하도록 하라.
- [6] LaTeX3 Project Team. *LaTeX 2_ε for authors*. `usrguide.tex`이라는 이름으로 LaTeX 2_ε 배포판에서 함께 제공하는 문서.
- [7] LaTeX3 Project Team. *LaTeX 2_ε for Class and Package writers*. `clsguide.tex`이라는 이름으로 LaTeX 2_ε 배포판에서 함께 제공하는 문서.
- [8] LaTeX3 Project Team. *LaTeX 2_ε Font selection*. `fntguide.tex`이라는 이름으로 LaTeX 2_ε 배포판에서 함께 제공하는 문서.
- [9] D. P. Carlisle. *Packages in the 'graphics' bundle*. 'graphics' 패키지 모음에 포함되어 있는 `grfguide.tex` 문서로서, LaTeX 배포판과 함께 제공됨.

- [10] Rainer Schöpf, Bernd Raichle, and Chris Rowley. *A New Implementation of L^AT_EX's verbatim Environments*. ‘tools’ 패키지 모음과 함께 제공되는 `verbatim.dtx` 문서. L^AT_EX 배포판과 함께 제공됨.
- [11] Vladimir Volovich, Werner Lemberg and L^AT_EX3 Project Team. *Cyrillic languages support in L^AT_EX*. `cyrguide.tex`이라는 이름으로 L^AT_EX 2_ε 배포판에서 함께 제공하는 문서.
- [12] Graham Williams. *The TeX Catalogue* 여러 T_EX 및 L^AT_EX 관련 패키지에 대한 완전한 목록. [CTAN:/tex-archive/help/Catalogue/catalogue.html](http://ctan.org/tex-archive/help/Catalogue/catalogue.html)에서 찾아볼 수 있음.
- [13] Keith Reckdahl. *Using EPS Graphics in L^AT_EX 2_ε Documents*, 이 문서에는 EPS 파일에 관한 내용과 L^AT_EX에서 EPS 그림의 사용법에 관한 내용이 기대보다 훨씬 많이 담겨 있다. [CTAN:/tex-archive/info/epslatex.ps](http://ctan.org/tex-archive/info/epslatex.ps)에서 찾을 수 있음.
- [14] Kristoffer H. Rose. *X_y-pic User's Guide*. X_y-pic 배포판과 함께 CTAN에서 내려받을 수 있음.
- [15] John D. Hobby. *A User's Manual for MetaPost*. <http://cm.bell-labs.com/who/hobby/>에서 내려받을 수 있음.
- [16] Alan Hoenig. *T_EX Unbound*. Oxford University Press, 1998, ISBN 0-19-509685-1; 0-19-509686-X (pbk.)
- [17] Urs Oswald. *Graphics in L^AT_EX 2_ε*, (임의의 원과 타원을 `picture` 환경 안에서 생성하는 Java 소스 파일을 포함하고 있음), *MetaPost — A Tutorial*. 두 문서를 <http://www.ursoswald.ch>에서 내려받을 수 있음.

찾아보기

- " , 21
- "' , 34
- "- , 34
- "--- , 34
- "< , 34
- "= , 34
- "> , 34
- "‘ , 34
- \$, 49
- \(, 49
- \) , 49
- \, , 51, 56
- , 22
- , 22
- \- , 20
- , 22
- , 22
- ., space after, 35
- ..., 24
- \: , 56
- \; , 56
- \@ , 35
- \[, 50
- \\ , 19, 40, 42, 118
- * , 19
- \] , 50
- ~ , 35
- A4 paper, 11
- A5 paper, 11
- å , 25
- abstract, 41
- Acrobat Reader, 79
- acute, 25
- \addtolength, 120
- æ , 25
- aeguill, 80
- \Alph, 34
- \alph, 34
- amsbsy, 61
- amsmath, 50, 51, 54–56, 58, 59, 61
- amssymb, 51, 63
- \and, 37
- ansinew, 27
- \appendix, 36, 37
- applemac, 27
- \ar, 104
- \arccos, 53
- \arcsin, 53
- \arctan, 53
- \arg, 53
- array, 56, 57
- article 클래스, 10
- \Asbuk, 34
- \asbuk, 34
- \author, 36, 85
- B5 paper, 11
- babel, 20, 21, 26, 33, 34
- \backmatter, 37
- \backslash, 6

- base font size, 11
- \baselineskip, 115
- beamer, 30, 90, 91
- \begin, 39, 94, 103
- \bibitem, 73, 74
- bibliography, 73
- \Big, 55
- \big, 55
- \Bigg, 55
- \bigg, 55
- \biggl, 59
- \biggr, 59
- \bigskip, 118
- \binom, 54
- block, 92
- bm, 61
- \bmod, 53
- bold face, 112
- bold symbols, 51, 61
- \boldmath, 61
- \boldsymbol, 61
- book 클래스, 10
- brazilian, 31
- calc, 120
- \caption, 46, 47
- \cdot, 53
- \cdots, 55
- Center, 40
- \chapter, 36
- \chaptermark, 76
- \ci, 107
- \circle, 97
- \circle*, 97
- \cite, 73, 74
- CJK, 29
- \cleardoublepage, 46
- \clearpage, 46
- \cline, 42
- color, 82, 87, 90
- comment, 7
- comment, 7
- comments, 7
- \cos, 53
- \cosh, 53
- \cot, 53
- \coth, 53
- cp1251, 27
- cp850, 27
- cp866nav, 27
- \csc, 53
- \date, 36
- dcolumn, 43
- \ddots, 55
- decimal alignment, 43
- \deg, 53
- \depth, 122, 123
- description, 39
- \det, 53
- Deutsch, 32
- dhucs, 28-30, 88
- \dim, 53
- displaymath, 50
- \displaystyle, 59
- doc, 13
- document font size, 11
- \documentclass, 10, 11, 14, 20
- dotless i and j, 25
- double sided, 11
- \dq, 33
- \dum, 107
- eepic, 93, 97
- \emph, 38, 112
- empty, 12
- Encapsulated POSTSCRIPT, 71, 82
- encodings

- font
 - LGR, 28
 - OT1, 28
 - T1, 28, 33
 - T2*, 33
 - T2A, 28, 33
 - T2B, 28
 - T2C, 28
 - X2, 28
- input
 - ansinew, 27
 - applemac, 27
 - cp1251, 27
 - cp850, 27
 - cp866nav, 27
 - koi8-ru, 27, 33
 - latin1, 27
 - macukr, 27
 - utf8, 27
- \end, 39, 94
- \enumBul, 34
- \enumEng, 34
- enumerate, 39
- \enumLat, 34
- epic, 93
- eqnarray, 57
- \eqref, 50
- equation, 50
- \EUR, 24
- europs, 23
- eurosym, 23
- executive paper, 11
- \exp, 53
- exponent, 52
- exscale, 13, 55
- extension
 - .aux, 14
 - .cls, 14
 - .dtx, 14
 - .dvi, 14, 72
 - .eps, 72
 - .fd, 14
 - .idx, 14, 75
 - .ilg, 14
 - .ind, 14, 75
 - .ins, 14
 - .lof, 14
 - .log, 14
 - .lot, 14
 - .sty, 12, 77
 - .tex, 9, 12
 - .toc, 14
- fancyhdr, 75, 76
- \fbox, 21
- Figure, 44
- figure, 45
- \flq, 33
- \flqq, 33
- Flushleft, 40
- flushright, 40
- foiltex, 10
- \foldera, 101
- \folderb, 101
- font
 - \footnotesize, 112
 - \Huge, 112
 - \huge, 112
 - \LARGE, 112
 - \Large, 112
 - \large, 112
 - \mathbf, 113
 - \mathcal, 113
 - \mathit, 113
 - \mathnormal, 113
 - \mathrm, 113
 - \mathsf, 113

- `\mathtt`, 113
- `\normalsize`, 112
- `\scriptsize`, 112
- `\small`, 112
- `\textbf`, 112
- `\textit`, 112
- `\textmd`, 112
- `\textnormal`, 112
- `\textrm`, 112
- `\textsc`, 112
- `\textsf`, 112
- `\textsl`, 112
- `\texttt`, 112
- `\textup`, 112
- `\tiny`, 112
- font encodings, 27
 - LGR, 28
 - OT1, 28
 - T1, 28, 33
 - T2*, 33
 - T2A, 28, 33
 - T2B, 28
 - T2C, 28
 - X2, 28
- font size, 112
- fontenc, 13, 27, 33
- `\footnote`, 38, 47
- `\footnotemark`, 38
- `\footnotesize`, 112
- `\footnotetext`, 38
- `\footskip`, 119
- formulae, 49
- `\frac`, 53
- frame, 92
- `\framebox`, 123
- `\frenchspacing`, 33, 35
- `\frontmatter`, 37
- `\frq`, 33
- `\frqq`, 33
- `\fussy`, 20
- `\gcd`, 53
 - geometry, 77
 - GhostScript, 71
 - graphicx, 71, 82, 87, 89, 90
 - grave, 25
- HE^AT_EX, 28–30, 87, 132
- `\headheight`, 119
 - textttheadings, 12
- `\headsep`, 119
- `\height`, 122, 123
- `\hline`, 42
- `\hom`, 53
 - horizontal
 - dots, 55
 - line, 52
 - space, 117
- `\href`, 84, 85, 87
- `\hspace`, 109, 117
- `\Huge`, 112
- `\huge`, 112
 - hyperref, 22, 79, 82, 83, 87–90
 - hypertext, 78
 - hyphenat, 77
- `\hyphenation`, 20
- `\idotsint`, 56
 - ifpdf, 86
- `\ifpdf`, 86
 - ifthen, 13
- `\ignorespaces`, 110
- `\ignorespacesafterend`, 110
- `\iiiint`, 56
- `\iiint`, 56
- `\iint`, 56
- `\include`, 15
- `\includegraphics`, 72, 82, 86, 122

- `\includeonly`, 15
- `\indent`, 116
 - `indentfirst`, 116
 - `index`, 74
- `\index`, 74, 75
- `\inf`, 53
- `\input`, 15, 111
 - input encodings
 - `ansinew`, 27
 - `applemac`, 27
 - `cp1251`, 27
 - `cp850`, 27
 - `cp866nav`, 27
 - `koi8-ru`, 27, 33
 - `latin1`, 27
 - `macukr`, 27
 - `utf8`, 27
 - `inputenc`, 13, 27, 33
- `\int`, 54
 - `international`, 26
 - `italic`, 112
 - `Itemize`, 39
- `\ker`, 53
 - Knuth, Donald E., 1
 - `koi8-ru`, 27, 33
- `\label`, 37, 50
 - Lamport, Leslie, 2
- `\LARGE`, 112
- `\Large`, 112
- `\large`, 112
 - LaTeX의 장점, 3
- `\LaTeX`, 21
 - LaTeX 2.09, 2
 - LaTeX_{2 ϵ} , 2
 - LaTeX3, 2, 4
- `\LaTeXe`, 21
 - `latexsym`, 13
 - `latin1`, 27
 - `layout`, 118
- `\ldots`, 24, 55
- `\left`, 55
 - `left alighted`, 40
- `\leftmark`, 76
 - `legal paper`, 11
 - `letter paper`, 11
- `\lg`, 53
 - `LGR`, 28
 - `ligature`, 24
- `\lim`, 53
- `\liminf`, 53
- `\limsup`, 53
- `\line`, 96, 101
- `\linebreak`, 19
- `\linespread`, 115
- `\linethickness`, 98, 99, 101
- `\listoffigures`, 46
- `\listoftables`, 46
- `\ln`, 53
- `\log`, 53
 - `long equations`, 57
 - `longtable`, 44
 - `lscommand`, 107
- `macukr`, 27
- `\mainmatter`, 37, 85
- `\makebox`, 122, 123
 - `makeidx`, 13, 74
 - `makeidx package`, 74
- `\makeindex`, 74
 - `makeindex program`, 74
- `\maketitle`, 36
- `\marginparpush`, 119
- `\marginparsep`, 119
- `\marginparwidth`, 119
- `\markleft`, 76
- `marvosym`, 24

- math, 49
- math font size, 59
- math spacing, 56
- `\mathbb`, 51
- `\mathbf`, 113
- `\mathcal`, 113
- mathematical
 - accents, 52
 - delimiter, 55
 - functions, 53
 - minus, 22
- `\mathit`, 113
- `\mathnormal`, 113
- `\mathrm`, 59, 113
- `\mathsf`, 69
- `\mathsf`, 113
- `\mathtt`, 113
- `\max`, 53
- `\mbox`, 21, 25, 51, 122
- METAPOST, 82
- `\min`, 53
- minimal 클래스, 10
- minipage, 122
- Mittelbach, Frank, 2
- mltex, 81
- mltex, 81
- `\multicolumn`, 43
- `\multirow`, 95, 98, 99
- `\newcommand`, 108, 109
- `\newenvironment`, 109
- `\newlength`, 121
- `\newline`, 19
- `\newpage`, 19
- `\newsavebox`, 100
- `\newtheorem`, 60
- `\noindent`, 116
- `\nolinebreak`, 19
- `\nonumber`, 58
- `\nopagebreak`, 19
- `\normalsize`, 112
- `\not`, 64
- `\oddsidemargin`, 119
 - œ, 25
 - one column, 11
 - option, 10
 - OT1, 28
- `\oval`, 99, 101
- `\overbrace`, 52
 - overfull hbox, 20
- `\overleftarrow`, 53
- `\overline`, 52
- `\overrightarrow`, 53
- package, 11
- packages
 - aeguill, 80
 - page style, 12
 - empty, 12
 - headings, 12
 - plain, 12
- `\pagebreak`, 19
- `\pageref`, 37, 78
- `\pagestyle`, 12
 - paper size, 11, 79
- `\paperheight`, 119
- `\paperwidth`, 119
- `\par`, 113
- `\paragraph`, 35
- `\parbox`, 122
 - parbox, 122
- `\parindent`, 116, 121
- `\parskip`, 116
- `\part`, 35, 36
- PDF, 78
- PDFLATEX, 90

- pdfL^AT_EX, 80, 90
- pdfL^AT_EX, 79
- pdfT_EX, 79
- pdftricks, 90
- \phantom, 47, 58
- picture, 93, 94, 97, 98
- plain, 12
- \pmb, 61
- \pmod, 53
- Português, 31
- POSTSCRIPT, 4, 9, 29, 46, 71, 72, 80–82, 89, 90, 94
 - Encapsulated, 71, 82
- ppower4, 90
- \Pr, 53
- preamble, 8
- \printindex, 75
- proc 클래스, 10
- \prod, 54
- prosper, 90
- \protect, 47
- \providecommand, 109
- \ProvidesPackage, 111
- pspicture, 94
- pstricks, 90, 93, 94, 97
- \put, 95–100
- pxfonts, 81
- \qbezier, 93, 95, 102
- \qqquad, 51, 56
- \quad, 51, 56
- quotation, 40
- Quote, 40
- \raisebox, 123
- \ref, 37, 50, 78
- \renewcommand, 108, 109
- \renewenvironment, 109
- report 클래스, 10
- \right, 55, 57
- \right., 55
- \rightmark, 76
 - roman, 112
- \rule, 109, 124
 - sans serif, 112
- \savebox, 100
 - Scandinavian letters, 25
- \scriptscriptstyle, 59
- \scriptsize, 112
- \scriptstyle, 59
- \sec, 53
- \section, 35, 47
- \sectionmark, 76
- \selectlanguage, 26
- \setlength, 94, 116, 120
 - setspace, 115
- \settodepth, 121
- \settoheight, 121
- \settowidth, 121
 - showidx, 75
- \sin, 53
 - single sided, 11
- \sinh, 53
- slanted, 112
- slides 클래스, 10
- \sloppy, 20
- \small, 112
 - Small Caps, 112
- \smallskip, 118
- \sqrt, 52
- \stackrel, 54
- \stretch, 109, 117
 - strut, 124
 - subarray, 54
- \subparagraph, 35
 - subscript, 52
- \subsection, 35

`\subsectionmark`, 76
`\substack`, 54
`\subsubsection`, 35
`\sum`, 54
`\sup`, 53
superscript, 54
syntonly, 13, 15

T1, 28, 33
T2*, 33
T2A, 28, 33
T2B, 28
T2C, 28
table, 44, 45
`\tableofcontents`, 36
tabular, 42, 44, 122
`\tan`, 53
`\tanh`, 53
`\TeX`, 21
`\texorpdfstring`, 86
`\text`, 51
`\textbackslash`, 6
`\textbf`, 112
`\textcelsius`, 23
textcomp, 23
`\texteuro`, 23
`\textheight`, 119
`\textit`, 112
`\textmd`, 112
`\textnormal`, 112
`\textrm`, 59, 112
`\textsc`, 112
`\textsf`, 112
`\textsl`, 112
`\textstyle`, 59
`\texttt`, 112
`\textup`, 112
`\textwidth`, 119
thebibliography, 73

`\thicklines`, 96, 99, 101
`\thinlines`, 99, 101
`\thispagestyle`, 12
tilde, 22
tilde (~), 35
`\tiny`, 112
title, 11
`\title`, 36
`\tnss`, 108
`\today`, 21
`\topmargin`, 119
`\totalheight`, 122, 123
two column, 11
txfonts, 81

ucs, 27, 29
umlaut, 25
`\underbrace`, 52
underfull hbox, 20
`\underline`, 38, 52
unicode, 29
`\unitlength`, 94, 96
units, 118
upright, 112
URL, 22
url, 22
`\url`, 22
`\usebox`, 100
`\usepackage`, 12, 23, 26, 27, 111
utf8, 27

`\vdots`, 55
`\vec`, 53
`\vector`, 96
vectors, 53
`\verb`, 41, 42
verbatim, 7, 76, 77
verbatim, 41, 76, 77
`\verbatim*`, 42

\verbatiminput, 77
 verse, 40
 vertical dots, 55
 \vspace, 117

 whitespace
 after commands, 6
 at the start of a line, 5
 \widehat, 52
 \widetilde, 52
 \width, 122, 123
 Word, 75
 www, 22
 WYSIWYG, 2, 3

 x2, 28
 Xpdf, 79
 xy, 103
 \xymatrix, 104

【가】

강세 부호, 25
 곱 기호, 54
 공백문자, 5
 명령 다음, 6
 괄호, 54
 구조, 7
 그리스 소문자, 52
 그림, 11, 71
 글꼴, 111
 글꼴 크기, 111
 기본 글꼴 크기, 11

【다】

단락, 17
 단면, 11
 단위, 117
 대괄호, 6
 대시, 22
 도 기호, 23

독일어, 26, 32

 따옴표, 21
 떠다니는 개체, 44

【마】

마침표, 24
 명령, 6
 \,, 49
 \), 49
 \,, 51, 56
 \-, 20
 \:, 56
 \;, 56
 \@, 35
 \[, 50
 \\, 19, 40, 42, 118
 *, 19
 \], 50
 \addtolength, 120
 \Alph, 34
 \alph, 34
 \and, 37
 \appendix, 36, 37
 \ar, 104
 \arccos, 53
 \arcsin, 53
 \arctan, 53
 \arg, 53
 \Asbuk, 34
 \asbuk, 34
 \author, 36, 85
 \backmatter, 37
 \backslash, 6
 \baselineskip, 115
 \begin, 39, 94, 103
 \bibitem, 73, 74
 \Big, 55
 \big, 55

<code>\Bigg</code> , 55	<code>\enumBul</code> , 34
<code>\bigg</code> , 55	<code>\enumEng</code> , 34
<code>\biggl</code> , 59	<code>\enumLat</code> , 34
<code>\biggr</code> , 59	<code>\eqref</code> , 50
<code>\bigskip</code> , 118	<code>\EUR</code> , 24
<code>\binom</code> , 54	<code>\exp</code> , 53
<code>\bmod</code> , 53	<code>\fbox</code> , 21
<code>\boldmath</code> , 61	<code>\flq</code> , 33
<code>\boldsymbol</code> , 61	<code>\flqq</code> , 33
<code>\caption</code> , 46, 47	<code>\foldera</code> , 101
<code>\cdot</code> , 53	<code>\folderb</code> , 101
<code>\cdots</code> , 55	<code>\footnote</code> , 38, 47
<code>\chapter</code> , 36	<code>\footnotemark</code> , 38
<code>\chaptermark</code> , 76	<code>\footnotetext</code> , 38
<code>\ci</code> , 107	<code>\footskip</code> , 119
<code>\circle</code> , 97	<code>\frac</code> , 53
<code>\circle*</code> , 97	<code>\framebox</code> , 123
<code>\cite</code> , 73, 74	<code>\frenchspacing</code> , 33, 35
<code>\cleardoublepage</code> , 46	<code>\frontmatter</code> , 37
<code>\clearpage</code> , 46	<code>\frq</code> , 33
<code>\cline</code> , 42	<code>\frqq</code> , 33
<code>\cos</code> , 53	<code>\fussy</code> , 20
<code>\cosh</code> , 53	<code>\gcd</code> , 53
<code>\cot</code> , 53	<code>\headheight</code> , 119
<code>\coth</code> , 53	<code>\headsep</code> , 119
<code>\csc</code> , 53	<code>\height</code> , 122, 123
<code>\date</code> , 36	<code>\hline</code> , 42
<code>\ddots</code> , 55	<code>\hom</code> , 53
<code>\deg</code> , 53	<code>\href</code> , 84, 85, 87
<code>\depth</code> , 122, 123	<code>\hspace</code> , 109, 117
<code>\det</code> , 53	<code>\hyphenation</code> , 20
<code>\dim</code> , 53	<code>\idotsint</code> , 56
<code>\displaystyle</code> , 59	<code>\ifpdf</code> , 86
<code>\documentclass</code> , 10, 11, 14, 20	<code>\ignorespaces</code> , 110
<code>\dq</code> , 33	<code>\ignorespacesafterend</code> , 110
<code>\dum</code> , 107	<code>\iiiint</code> , 56
<code>\emph</code> , 38, 112	<code>\iiint</code> , 56
<code>\end</code> , 39, 94	<code>\iint</code> , 56

<code>\include</code> , 15	<code>\mbox</code> , 21, 25, 51, 122
<code>\includegraphics</code> , 72, 82, 86, 122	<code>\min</code> , 53
<code>\includeonly</code> , 15	<code>\multicolumn</code> , 43
<code>\indent</code> , 116	<code>\multirow</code> , 95, 98, 99
<code>\index</code> , 74, 75	<code>\newcommand</code> , 108, 109
<code>\inf</code> , 53	<code>\newenvironment</code> , 109
<code>\input</code> , 15, 111	<code>\newlength</code> , 121
<code>\int</code> , 54	<code>\newline</code> , 19
<code>\ker</code> , 53	<code>\newpage</code> , 19
<code>\label</code> , 37, 50	<code>\newsavebox</code> , 100
<code>\LaTeX</code> , 21	<code>\newtheorem</code> , 60
<code>\LaTeXe</code> , 21	<code>\noindent</code> , 116
<code>\ldots</code> , 24, 55	<code>\nolinebreak</code> , 19
<code>\left</code> , 55	<code>\nonumber</code> , 58
<code>\leftmark</code> , 76	<code>\nopagebreak</code> , 19
<code>\lg</code> , 53	<code>\not</code> , 64
<code>\lim</code> , 53	<code>\oddsidemargin</code> , 119
<code>\liminf</code> , 53	<code>\oval</code> , 99, 101
<code>\limsup</code> , 53	<code>\overbrace</code> , 52
<code>\line</code> , 96, 101	<code>\overleftarrow</code> , 53
<code>\linebreak</code> , 19	<code>\overline</code> , 52
<code>\linespread</code> , 115	<code>\overrightarrow</code> , 53
<code>\linethickness</code> , 98, 99, 101	<code>\pagebreak</code> , 19
<code>\listoffigures</code> , 46	<code>\pageref</code> , 37, 78
<code>\listoftables</code> , 46	<code>\pagestyle</code> , 12
<code>\ln</code> , 53	<code>\paperheight</code> , 119
<code>\log</code> , 53	<code>\paperwidth</code> , 119
<code>\mainmatter</code> , 37, 85	<code>\par</code> , 113
<code>\makebox</code> , 122, 123	<code>\paragraph</code> , 35
<code>\makeindex</code> , 74	<code>\parbox</code> , 122
<code>\maketitle</code> , 36	<code>\parindent</code> , 116, 121
<code>\marginparpush</code> , 119	<code>\parskip</code> , 116
<code>\marginparsep</code> , 119	<code>\part</code> , 35, 36
<code>\marginparwidth</code> , 119	<code>\phantom</code> , 47, 58
<code>\markleft</code> , 76	<code>\pmb</code> , 61
<code>\mathbb</code> , 51	<code>\pmod</code> , 53
<code>\mathrm</code> , 59	<code>\Pr</code> , 53
<code>\max</code> , 53	<code>\printindex</code> , 75

<code>\prod</code> , 54	<code>\subsubsection</code> , 35
<code>\protect</code> , 47	<code>\sum</code> , 54
<code>\providecommand</code> , 109	<code>\sup</code> , 53
<code>\ProvidesPackage</code> , 111	<code>\tableofcontents</code> , 36
<code>\put</code> , 95–100	<code>\tan</code> , 53
<code>\qbezier</code> , 93, 95, 102	<code>\tanh</code> , 53
<code>\qquad</code> , 51, 56	<code>\TeX</code> , 21
<code>\quad</code> , 51, 56	<code>\texorpdfstring</code> , 86
<code>\raisebox</code> , 123	<code>\text</code> , 51
<code>\ref</code> , 37, 50, 78	<code>\textbackslash</code> , 6
<code>\renewcommand</code> , 108, 109	<code>\textcelsius</code> , 23
<code>\renewenvironment</code> , 109	<code>\texteuro</code> , 23
<code>\right</code> , 55, 57	<code>\textheight</code> , 119
<code>\right.</code> , 55	<code>\textrm</code> , 59
<code>\rightmark</code> , 76	<code>\textstyle</code> , 59
<code>\rule</code> , 109, 124	<code>\textwidth</code> , 119
<code>\savebox</code> , 100	<code>\thicklines</code> , 96, 99, 101
<code>\scriptscriptstyle</code> , 59	<code>\thinline</code> , 99, 101
<code>\scriptstyle</code> , 59	<code>\thispagestyle</code> , 12
<code>\sec</code> , 53	<code>\title</code> , 36
<code>\section</code> , 35, 47	<code>\tnss</code> , 108
<code>\sectionmark</code> , 76	<code>\today</code> , 21
<code>\selectlanguage</code> , 26	<code>\topmargin</code> , 119
<code>\setlength</code> , 94, 116, 120	<code>\totalheight</code> , 122, 123
<code>\settodepth</code> , 121	<code>\underbrace</code> , 52
<code>\settoheight</code> , 121	<code>\underline</code> , 38, 52
<code>\settowidth</code> , 121	<code>\unitlength</code> , 94, 96
<code>\sin</code> , 53	<code>\url</code> , 22
<code>\sinh</code> , 53	<code>\usebox</code> , 100
<code>\sloppy</code> , 20	<code>\usepackage</code> , 12, 23, 26, 27, 111
<code>\smallskip</code> , 118	<code>\vdots</code> , 55
<code>\sqrt</code> , 52	<code>\vec</code> , 53
<code>\stackrel</code> , 54	<code>\vector</code> , 96
<code>\stretch</code> , 109, 117	<code>\verb</code> , 41, 42
<code>\subparagraph</code> , 35	<code>\verbatim*</code> , 42
<code>\subsection</code> , 35	<code>\verbatiminput</code> , 77
<code>\subsectionmark</code> , 76	<code>\vspace</code> , 117
<code>\substack</code> , 54	<code>\widehat</code> , 52

\widetilde, 52
 \width, 122, 123
 \xymatrix, 104
 모듈로 함수, 53
 문서 기본 글꼴, 11
 물결표, 52

【바】

백슬래시, 6
 벡터, 53
 분수, 53
 블랙보드 볼드, 51

빼기 부호, 22

【사】

상호 참조, 37
 소수점 정렬, 43
 수식군, 57
 수직 간격, 117
 수평
 간격, 117
 수학 짝맞춤 기호, 55
 수학식, 49
 쉼표, 24
 스페이스, 5

【아】

양면, 11
 엔대시, 22
 엠대시, 22
 옵션 인자, 6
 용지 크기, 79
 위치 지정자, 44
 유보된 문자들, 5
 2단, 11
 인자(parameter), 6
 1단, 11
 입력 파일, 8

【자】

적분 기호, 54
 전처리부, 8
 점 세 개, 55
 점 세 개를 대각선으로, 55
 제곱근, 52
 주석, 7
 줄 간격, 115
 더블스페이스, 115
 줄바꿈, 19
 줄임표, 24

짝맞춤 문자, 54
 쪽 양식, 12

【차】

차례, 36
 참고문헌 목록, 73
 찾아보기, 74
 채색 글씨, 11

【카】

캡슐화된 포스트스크립트, 71, 82
 크누쓰, 1

【타】

특수 문자, 25

【파】

파일 유형, 12
 패키지, 7, 11, 107
 aeguill, 80
 amsbsy, 61
 amsfonts, 51, 69
 amsmath, 50, 51, 54–56, 58, 59,
 61
 amssymb, 51, 63
 babel, 20, 21, 26, 33, 34
 beamer, 30, 90, 91
 bm, 61

- calc, 120
- CJK, 29
- color, 82, 87, 90
- comment, 7
- dcolumn, 43
- dhucs, 28–30, 88
- doc, 13
- eepic, 93, 97
- epic, 93
- europs, 23
- eurosym, 23
- exscale, 13, 55
- fancyhdr, 75, 76
- fontenc, 13, 27, 33
- geometry, 77
- graphicx, 71, 82, 87, 89, 90
- hyperref, 22, 79, 82, 83, 87–90
- hyphenat, 77
- ifpdf, 86
- ifthen, 13
- indentfirst, 116
- inputenc, 13, 27, 33
- latexsym, 13
- layout, 118
- makeidx, 74
- makeidx, 13, 74
- marvosym, 24
- mathrsfs, 69
- mathtext, 33
- mltex, 81
- pdftricks, 90
- ppower4, 90
- prosper, 90
- pstricks, 90, 93, 94, 97
- pxfonts, 81
- setspace, 115
- showidx, 75
- syntonly, 13, 15
- textcomp, 23
- txfonts, 81
- ucs, 27, 29
- unicode, 29
- url, 22
- verbatim, 7, 76, 77
- xy, 103
- 페이지 레이아웃, 118
- 포르투갈어, 31
- 폰트 인코딩, 13
- 표, 42
- 표지, 11
- 폴리는 명령, 47
- 프랑스어, 31
- 프로그램
 - makeindex, 74
- 【하】**
 - 하이퍼텍스트, 78
 - 하이픈, 22
 - 합 기호, 54
 - 합자, 24
 - 화살표 부호, 53
 - 확장명, 12
 - 환경
 - Center, 40
 - Figure, 44
 - Flushleft, 40
 - Itemize, 39
 - Quote, 40
 - abstract, 41
 - array, 56, 57
 - block, 92
 - comment, 7
 - description, 39
 - displaymath, 50
 - enumerate, 39
 - eqnarray, 57
 - equation, 50

figure, 45
flushright, 40
frame, 92
longtable, 44
lscommand, 107
math, 49
minipage, 122
parbox, 122
picture, 93, 94, 97, 98
pspicture, 94
quotation, 40
subarray, 54
table, 44, 45
tabular, 42, 44, 122
thebibliography, 73
verbatim, 41, 76, 77
verse, 40

역자후기 (v. 4.17)

KTUG이 생기기 전에 필자의 개인 홈페이지에서 처음 번역이 이루어진 lshort-kr은 그 후 CTAN에 등록되어 우리글로 된 L^AT_EX 입문서로서 많은 분들에게 읽혀왔다. 지금 돌이켜보면 당시 H^AL^AT_EX을 기반으로 번역 작업을 하면서 고생했던 것은 영문을 옮기는 것도 옮기는 것이지만 L^AT_EX에서 한글을 구현하는 것 자체가 쉽지 않았던 면이 있었다.

KTUG이 이루어놓은 놀라운 업적들이 이 번역본에 고스란히 담겨 있다. 우선 최종 출력물인 PDF의 품위 자체가 달라졌다. 한글 책갈피, 텍스트의 추출 · 검색, 하이퍼링크 등, 3.20판을 번역할 당시에는 잘 상상하기 어렵던 일들이 너무나 손쉽게 가능해졌다. 또한 유니코드 기반의 한글로 이행함으로써 더이상 H^AL^AT_EX의 EUC-KR 한계를 걱정하지 않아도 되는 행복한 상황에서 4.17판을 번역하게 되었다. 이 한글판 lshort-kr은 unicode/dhucs 패키지를 이용하여 한글을 구현하였다.

기술적 뒷받침이 이루어졌으므로 이제는 내용의 질을 제고할 때라고 생각한다. 이번 번역도 자원한 분들과 함께 공동작업으로 이루어졌다. 영어에 능한 분들과 언어학을 전공하신 분까지 번역에 합류함으로써 이제는 번역의 질에 있어서도 부끄럽지 않을 정도가 되었다고 생각한다.

이번에 공동번역자들이 합의한 번역 원칙은 다음과 같다.

- 영문을 그대로 옮겨놓는 번역을 피하고 실제 L^AT_EX에 입문하는 초보자들이 쉽게 이해하고 적용할 수 있도록 문장을 완전히 새로 쓴다. 필요하다면 내용을 보충하거나 생략할 수 있다. 중요한 것은 입문서로서 이 책의 전달력이자 원문의 충실한 재현에 있지 않다. 즉, 번역문에서 어떤 영문 문장의 기미도 발견할 수 없도록, 아름다운 우리말로 이루어진 문서를 만드는 것이 목표이다. (우리말식 용어를 사용해야 한다는 뜻은 결코 아님).
- 예제는 이번에도 영문을 그대로 노출시킨다. 한글 L^AT_EX의 발전에 힘입어 예제를 한글화하는 것이 어려운 일은 아니지만 여전히 이 책은 L^AT_EX 사용법에 관한 책이다. 그러나 한글 L^AT_EX 사용환경이 현저히 다르거나 한글화와 관련한 중요한 사항이 있을 때는 이에 대한 언급을 별도의 절

이나 역주로 만들어 붙인다.

- 용어의 통일은 중요하다. 역어의 선택은 되도록이면 우리말화하되 우리 말 용어 자체를 지나치게 중시한 나머지 도무지 알 수 없거나 머리 속에서 영어로 다시 옮겨야 이해가 되는 것을 배제하고 적당한 우리말이 없는 경우에는 차라리 흔히 쓰이는 영어식 표현을 그대로 쓴다.

모든 경우에 이 원칙이 완전히 관철되지는 않았을 것이다. 그러나 번역자들이 지향하는 번역의 방향은 대체로 일치하였다고 생각한다.

한국어판이 개선되지 않는에도 불구하고 메일링 리스트에서 삭제하지 않고 꾸준히 훌륭한 문서를 만들어 온 저자 Tobias Oetiker 씨에게 감사한다. 원래 한국어판 번역본에 실렸던 글을 더 확장하고 보충하여 영문판에 실리도록 글을 써주신 김정식 님께도 감사드린다. 이 번역본에 새로이 수정하여 실린 “한국어 지원” 절은 lshort의 다음 판에서 이용할 수 있도록 Oetiker 씨에게 수정본을 보낼 생각이다. 사실상 KTUG의 활동을 가능케 한 조진환 박사, KTUG에 깊은 애정을 보이시는 지도자이자 후원자이신 남상호 박사, dhucs와 dhhangul의 저자로서 한글화에 중요한 기여를 하신 김도현 교수, KTUG의 중심인 이주호 님, 이호재 님, KTUG Collection을 가능하게 한 홍석호 님, 그밖에 KTUG과 lshort-kr에 관심과 격려를 보내주신 모든 질문자와 답변자분들께 감사드린다. 이주호 님은 번역본 전체를 읽으면서 주교정자로서의 역할을 담당해주셨다. 특별히 교정·교열 과정에서 긴 교정표를 작성해주신 이상직 교수, 그리고 번역본을 읽고 의견을 제시해주신 딸기아빠 님과 커썸 님께도 감사드린다. 번역본이 더 읽고 이해하기 쉬워진 것은 전적으로 이 분들의 공이다.

번역이 이루어질 무렵 한글날을 맞아 한겨레신문사에서 한겨레결체를 공개한 것은 공개 한글 글꼴이 부족한 \TeX 공동체에 좋은 선물이었다. 이 문서는 한겨레결체를 본문 글꼴로 채용하였다.

이 번역본에 잘못이 있다면 책임은 김강수에게 있다. 공동 역자들은 오역에 대하여 책임이 없다. 초보자분으로서 이 책에서 조금이라도 얻은 것이 있다면 역자들에게 알려주기 바란다. 개선을 위한 의견, 오류의 지적도 환영한다. KTUG 게시판이나 전자우편을 이용할 수 있다.

함께 고생한 공동 역자들의 노력에 감사하고 행운을 빌면서.

번역자 : 김강수, 이기황, MIKA, 샘처럼, 김지운.

공동번역자를 대표하여...

김강수 info@mail.ktug.or.kr

역자후기 (v. 3.20)

이 책은 사용법이 쉽지만은 않은 L^AT_EX의 입문서로 이미 정평이 있다.

당연히 L^AT_EX 2_ε를 제대로 이해하고 쓰기 위해서는 *The L^AT_EX Companion* [3]이 있어야 할 것이고, 좀더 고급의 독자들은 *The T_EXbook*을 보아야 하겠지만, 논문 작성 등 일반적 용도에는 이 책이 제공하는 정도의 기능만 숙지하더라도 충분히 자신의 목적을 달성할 수 있을 것이다.

이 책을 번역해야겠다는 생각은 오래 전부터 가지고 있었는데, 그것을 실행에 옮길 엄두를 내기가 어려웠다. 우선, 한글판 lshort가 과연 필요할 것인가도 확실하기 어려웠고(왜냐하면 어차피 이 책을 한국어로 옮긴다 하더라도 예제는 여전히 영어 예제를 쓸 수밖에 없으며, 한글 구현에 관한 사항은 이 글의 ‘번역’에서는 다룰 수 없었기 때문이다.), 사실 초창기 lshort는 영문판도 컴파일이 잘 되지 않는 경우가 있어서, 이것이 과연 한글로 제대로 동작할 것인지 확실할 수 없는 상태였기 때문에, 그냥 영문판을 보는 것으로 만족하고 지낸 것이 사실이다.

그러던 차에, 나의 개인 홈페이지¹ 게시판에서 이 문제를 제기했더니 강운배 · 장대훈 님이 흔쾌히 돕겠다는 의사를 밝혀 주셨다. 이렇게 의기투합하여, 대부분의 본문을 한글로 옮기는 일을 두 분이 하고, 나는 한글 L^AT_EX으로 컴파일이 되도록 맞추는 일을 주로 하면서 초벌번역이 이루어졌다. 초벌번역이 끝날 무렵, 김재우 님께서는 다른 경로로 나에게 연락을 해오셨는데, 3.1의 번역을 이미 해두신 적이 있다는 것이었다. 이렇게 전체의 번역이 이루어진 후, 내가 각 장을 다시 읽으면서 교열하고 오역을 수정하는 작업을 거쳐 마침내 한국어판 lshort를 출판(!)하게 되었다.

각 장별로 최종적으로 사용된 텍스트의 초벌 역자와 교열자는 표 8.1과 같다.

이 책이 L^AT_EX에 입문하는 분들에게 좋은 선물이 되기를 바란다. 사실 한글로 이루어진 T_EX 관련서적이 거의 없다 해도 좋을 정도의 상황에서, 이 글이 가치있는 입문서 구실을 충분히 할 것으로 믿는다.

번역상 주의한 것은 다음과 같다.

¹ <http://www.doeun.pe.kr>

감사의 말, 서문	강윤배·장대훈	김강수
제1장	김강수	장대훈·김강수
제2장	강윤배	장대훈·김강수
제3장	장대훈	김강수
제4장	김재우*	김강수
제5장	장대훈	김강수
참고문헌	장대훈	김강수

* 장대훈 님의 번역본도 있었는데, 김재우 님의 텍스트를 주로 살리면서 장대훈 님의 번역을 참조하여 교열하였음.

표 8.1: 각 장별 번역자와 교열자

- 예제들은 영문을 그냥 노출시켰다. 이렇게 한 이유는, 이 책이 $\text{\LaTeX} 2_{\epsilon}$ 에 대한 설명이지 \HLaTeX 에 대한 설명이 아니라는 점 때문이었다. 다시 말하면 이 예제들을 한글화했을 때, 그것은 $\text{\LaTeX} 2_{\epsilon}$ 를 통해 실행되는 것이 아니라 \HLaTeX 을 통해서 실행되는 것이므로, 이 책의 원래 의도와는 동떨어진 것이 된다. \LaTeX 에서의 한글 사용에 대한 좋은 입문서가 나오기를 바라는 마음 간절하다. 아니 그보다, 안심하고 쓸 수 있는 한글 \TeX 이 하나 있었으면 하는 생각도 든다.
- 문장의 번역은 무엇보다도 \LaTeX 입문자들이 가장 잘 이해할 수 있게 하는 데 초점을 맞추었다. 필요하다면 설명을 길게 덧붙이기도 했고 몇 가지 역자에 의한 보충도 추가하였다. 이런 시도가 도움이 되기를 바란다.
- 용어는 공동번역자들이 통일하기 위해서 여러번 시도했지만 완전히 일치하지는 못했을 수도 있을 것이다. 이 문제는 차차 고쳐가겠다.
- 최근 CTAN의 디렉토리 구조가 바뀌면서 이 책에 나오는 URL 정보가 달라진 것이 있어서 그것을 바로잡았다.

책을 옮기는 일은 솔직히 말하면 쉽지 않았다. 그 과정에서 격려해 준 김도현 님, 이현호 님, ‘무식인’ 님을 비롯한 모든 분들에게 특별히 감사의 말을 전한다. 초벌 번역본과 교열본을 읽고 어색한 표현을 지적해 준 \LaTeX 을 전혀 모르는 나의 학생들에게도 고맙다는 인사를 전한다. \HLaTeX 의 저자인 은광희 님께 감사한다. 한글 \LaTeX 이 없었으면 이 글의 번역은 불가능했을 것이다.

이 번역본의 모든 책임은 김강수에게 있다. 다른 공동역자들은 오역에 대하여 책임이 없다.

번역자 : 김강수(karnes@doeun.pe.kr), 강윤배(pear@postech.ac.kr), 장대훈(webmaster@texworld.pe.kr), 김재우(dbunix@kitinet.co.kr)