

워드 프로세서 사용자를 위한 \LaTeX version 1.0.6

Guido Gonzato, Ph.D. <gonzato@sci.univr.it>

Università di Verona (Italy)
Facoltà di Scienze MM. FF. NN.[†]

한글 \TeX 사용자 그룹(KTUG) 문서화 작업팀 옮김

2003년 12월 30일

요 약

\LaTeX 을 이용한 문서 처리가 워드 프로세서를 이용한 문서 작업보다 나은 점이 몇 가지 있다. 그러나, 특정한 결과를 얻으려면 어떤 작업을 수행해야 하는지를 초보자들이 잘 알기가 어렵다. 이 지침서는 워드 프로세서를 이용한 작업과 \LaTeX 을 이용한 조판을 비교함으로써 워드에서 \LaTeX 으로 옮겨가는 과정을 좀 쉽게 만들어보려고 작성되었다. 우선 워드 프로세서의 주요 기능들을 열거한 다음, 거기에 해당하는 기능을 \LaTeX 명령으로는 어떻게 얻는지 차례로 설명한다. 예제도 풍부하게 제공하겠다.

목 차

제 1 절 들어가는 말	1
1.1 기본적인 사항	2
1.1.1 에디터가 지원하는 기능들	3
1.1.2 패키지 추가	3
1.2 문서 작성의 황금률	5
제 2 절 파일(File) 메뉴	5
2.1 새 문서(File/New)	5
2.2 새 이름으로(File/Save As...)	6
2.3 서식파일로 저장(File/Save As Template)	6
2.4 다른 문서 가져오기(File/Import)	6
2.5 편집용지 설정(File/Page Setup)	7
2.5.1 머리글/바닥글(Page Setup/Headers and Footers)	8
2.6 프린터 설정(File/Printer Setup)	8
2.7 프린트 미리보기(File/Print Preview)	9

[†]Ca' Vignal II, Strada Le Grazie 15, 37134 Verona (Italy)

2.8 인쇄(File/Print)	9
2.9 버전 관리(File/Versions)	9
제 3 절 편집(Edit) 메뉴	9
3.1 자동완성(Edit/Autotext)	10
제 4 절 삽입(Insert) 메뉴	10
4.1 쪽나누기, 문단나누기(Insert/Breaks)	10
4.2 문단 번호(Insert/Enumerated List)	11
4.3 특수문자(Insert/Special Character)	12
4.3.1 유로(EURO)화 기호	13
4.4 수식(Insert/Formula)	13
4.5 각주(Insert/Footnote)	14
4.5.1 미주	14
4.6 목차 넣기(Insert/Indices)	15
4.7 수평 수직 간격(Insert/Vertical and Horizontal Space)	15
4.8 탭(Insert/Tabs)	16
4.9 상호참조(Insert/Cross Reference)	16
4.10 난외주(Insert/Margin Notes)	17
4.11 글틀/글상자(Insert/Frame)	17
4.12 그림삽입(Insert/Figure)	17
4.12.1 그림과 표를 텍스트로 돌려짜기	19
4.13 도형삽입(Insert/Shapes)	20
4.14 선그리기(Insert/Line)	21
4.15 하이퍼링크 삽입(Insert/Hyperlink)	21
4.16 주석문(Insert/Comment)	22
제 5 절 모양(Format) 메뉴	22
5.1 행 간격(Format/Line Spacing)	23
5.2 글자모양(Format/Character)	23
5.2.1 밑줄 굵기	24
5.2.2 글자 크기(Format/Character Size)	25
5.2.3 글꼴 바꾸기(Format/Character Font)	25
5.2.4 글자 색(Format/Character Colour)	27
5.3 문단모양(Format/Paragraph)	27
5.3.1 문단의 가로 정렬(Format/Paragraph Horizontal Alignment)	28
5.3.2 문단의 간격 벌림(Format/Paragraph Vertical Alignment)	28
5.3.3 문단 여백(Format/Paragraph Margins)	29
5.3.4 문단 들여쓰기(Format/Paragraph Indentation)	30
5.4 문단 테두리와 음영(Format/Paragraph Border and Shade)	30
5.5 색상(Format/Colour)	31
5.6 다단편집(Format/Columns)	31
제 6 절 표(Table) 메뉴	32

6.1	행 사이에 임의의 간격 부여하기	34
6.1.1	소수점 정렬	35
6.2	slashbox로 셀에 선 긋기	35
6.3	TeX 표로 데이터 가져오기	36
제 7 절	도구(Tools) 메뉴	36
7.1	대량 메일 작성(Tools/Mail Merges)	36
7.2	레이블 만들기(Tools/Labels)	37
7.3	언어선택(Tools/Default Language)	39
7.4	자동 하이픈(Tools/Hyphenation)	40
7.5	맞춤법 검사(Tools/Spell Check)	40
제 8 절	도움말(Help) 메뉴	40
제 9 절	마치는 말	41
부록 A	문서마당(Document Templates)	42

표 목차

1	Emacs, Vim, Jed(IDE mode)의 기능별 키 설정	4
2	특수문자를 얻는 몇 가지 방법	12
3	글꼴 속성	23
4	글꼴 크기	24
5	글꼴 가족	26
6	표준 TeX 환경들	28
7	표 예제	33

그림 목차

1	이 글의 저자를 나타내는 스마일리	18
2	A Gnuplot graph	18
3	Book template	42
4	Report template	42
5	Letter template	43
6	How to write a notice	43
7	How to write a poster	44

역자의 말

이 문서는 가상의 워드 프로세서를 상정하고, 그 워드 프로세서에서 제공하는 기능을 \LaTeX 으로 어떻게 구현하는지 설명한 글이다. \LaTeX 을 사용하려는 사람 대부분이 워드 프로세서에 익숙한 경우가 많은 우리나라에서 이 글은 특히 유용할 것으로 생각되어서 번역하였다.

원문의 각주는 그대로 두고, 많은 역주를 붙였다. 역주는 로마 숫자로 매겼다. 역주의 목적은 원문의 불충분한 부분을 보충하거나, 역자의 의견을 달거나, 한글 환경에 맞지 않은 부분을 부연하거나, 윈도 환경에서 필요한 내용을 추가하거나 하기 위해서 붙인 것으로, 이 번역물의 가장 큰 특징을 이룬다.

예제는 되도록 원문을 살리려고 하였다. 한글 예제는 한글 \LaTeX (H \LaTeX)을 사용하여야 얻을 수 있다.

이 PDF 문서는 원문과 달리 기본 글꼴을 바꾸었다. 영문은 Utopia, 한글은 아시아 폰트 트루타입을 이용하였다. 이 번역 문서에 쓰인 아시아 폰트의 설정과 사용에 대해서는 KTUG(<http://www.ktug.or.kr>)에서 도움을 얻을 수 있을 것이다.

번역은 김강수가 하였고, 이주호는 번역된 글을 읽고 잘못된 곳을 지적하고 바로잡았다.

제 1 절 들어가는 말

먼저, 이 문서는 \LaTeX 입문서가 아니라는 점을 말해두려 한다. 이 글을 읽는 독자들은 최소한 \LaTeX 이 뭔지 들어본 적이 있고, 기본적인 명령도 대강 알고 있는 것으로 가정하겠다. 이 문서에서 내가 설명하려고 하는 것은 \LaTeX 이 과연 워드 프로세서를 대신할 가능성이 있는가 하는 것이다. \LaTeX 이 이런 저런 워드 프로세서보다 훨씬 더 낫다고 생각할 사람도 있을 것이다. 틀림없다. 많은 수학적식을 포함한 복잡한 문서를 저술하는 경우에는 특히 그렇다.

그러나, 워드 프로세서는 오늘날 사무 자동화 환경에서는 “킬러 프로그램”이다. 친숙한 WYSIWYG 인터페이스를 갖춘 워드 프로세서가 \LaTeX 보다 더 배우기 쉽다고 생각하는 사람이 많다. 그리고, 평범한 사무보조원들도 워드 프로세서 사용법은 비교적 짧은 시간 안에 익힐 수 있다. 문제는, 이 물건들이 시간이 갈수록 점점 느려지고, 덩치가 커지고,¹ 버그투성이가 되고, 비싸지고, 바이러스에 노출되는 데다가 서로 간에 호환가능하지도 않게 되어가고 있다는 것이다.

\LaTeX 이 좋은 해결책이다. 작성하려는 문서가 즉흥적이고 비구조적인 문서인 경우를 제외하고는, 사실상 \LaTeX 을 쓰는 것이 바람직한 방향이라 할 수 있다. 물론 plain \TeX 을 이용할 생각을 해볼 수도 있다. 그러나 그렇게 되면 \LaTeX 이 제공하는 많은 강력한 기능들을 사용하지 못하게 되는 셈이다.

요약하자. \LaTeX 을 쓰면서도, 이따금 워드 프로세서가 제공하는 기능을 사용해보고 싶을 때가 있을 것이다. 한때 좋아했던 :-) 워드 프로세서를 쓸 때 하던 것과 비슷한 효과를 \LaTeX 에서도 얻을 수 있는 방법을 알면 좋겠다는 생각이 들 때가 있을 것이다.

내가 이 간단한 지침서를 쓰게 된 이유가 바로 그것이다. 앞에서 말했듯이 \LaTeX 의 기본적인 것을 이미 이해하고 있는 사람에게 도움을 주려는 것이 이 글의 목적이다. 만약 \LaTeX 에 대해 잘 알지 못하는 경우라면, <http://www.tug.org/begin.html> 페이지^{*i}로 가서 “The (Not So) Short Introduction to $\text{\LaTeX}2_{\epsilon}$ ”^{**ii}나 “A Gentle Introduction to \TeX ” 같은 문서를 다운받아 읽어볼 것을 권장한다.

¹ 예전에 나는 나의 학위논문을 128k, Z80 프로세서를 갖춘 가정용 컴퓨터에서 작성했다. WordStar라는 워드 프로세서와 나의 논문 전부가 한 장의 CP/M 부팅가능한 720K 플로피 디스크에 들어가고도 여분의 공간이 있었다!

^{*i} 원문에는 <http://www.ctan.org/starter.html>로 되어 있으나, 역자가 URL을 올바르게 고쳤다.

^{**ii} 한국어판은 <http://www.doeun.pe.kr/frames/lshort.html>에서 구할 수 있다.

다음 절에서부터, 우리는 어떤 가상의 워드 프로세서^{*iii}의 메뉴와 메뉴 항목을 검토하면서 그것과 동일한 작업을 \LaTeX 으로 하는 방법을 찾아보려 한다. 당신이 \LaTeX 순정주의자로서 워드 프로세서와 비교하는 것 자체가 싫다면, 이런 식의 접근은 따분하기만 한 일... 공연히 마음 상하지 말고, 여기서 접어라.

1.1 기본적인 사항

워드 프로세서가 제공하는 기능 가운데 많은 부분은 우선 에디터에 의해서 실행할 수 있다. 나머지는 표준적인 \LaTeX 명령으로 구현된다. 그밖의 것들은 패키지를 사용하여 얻는다. 패키지라는 것은 새로운 명령어를 제공하는 \LaTeX 확장 매크로 집합이다. 주변에 다양한 패키지들이 있다. 유일한 문제는 뭐가 어디에 있는지, 무슨 일을 하는 것인지, 그것을 어떻게 인스톨할 것인지를 알아야 한다는 것이다. 패키지에 대해서 더 자세한 것은 제 1.1.2 절에서 다루겠다.

패키지 및 그밖의 \TeX 관련 자료들은 CTAN(The Comprehensive \TeX Archive Network)을 구성하는 여러 사이트에서 찾아볼 수 있다. 앞에서 말한 <http://www.ctan.org>가 그것이다. 이곳에서 미리 사이트들의 목록도 찾을 수 있을 것이다. 이후부터, CTAN:이라 하면, 'CTAN 미리 사이트의 \TeX 디렉토리'를 가리키는 것으로 하겠다. 예를 들어 자신의 플랫폼에 알맞은 \LaTeX 시스템은 [CTAN://systems](http://www.ctan.org/systems)에서, 즉, CTAN 미리 사이트 가운데 이를테면 <http://www.tex.ac.uk/tex-archive/systems/>에서 가져올 수 있다.

또다른 문제는 이런 것이다. \LaTeX 이 텍스트 조판(typesetting)을 해준다고 치자. 그렇다면 글은 무엇으로 써야 하는가? 답: 좋은 에디터를 이용하면 된다. \LaTeX 을 이해하는 에디터라면 더 좋다. \LaTeX 원본(source) 작성을 위한 에디터도 있는데, 그 중 일부는 WYSIWYG이거나 거의 WYSIWYG에 가까운 것도 있다.

플레인 텍스트를 저장할 수 있는 에디터라면 어떤 것도 괜찮기는 하지만(심지어 Windows의 메모장(notepad)도 된다), 많은 \TeX 전문가들(\TeX nicians)은 아래와 같은 것을 설치하고 이용할 것을 권장한다.

- GNU Emacs (<http://www.gnu.org/software/emacs/emacs.html>) 또는 Xemacs (<http://www.xemacs.org>)와 AU \TeX (CTAN://support/auctex).
- vi나 그 클론 가운데 하나인 Vim (<http://www.vim.org>). 가장 많이 쓰이는 것이다.
- 저자의 개인적인 취향은 Jed라는 것인데, (<http://space.mit.edu/~davis/jed/>), 다른 에디터를 흉내낼 수 있는 축소판 Emacs라고 할 수 있다. CTAN://support/jed에 저자가 만든 jed의 \LaTeX 모드 지원을 찾아볼 수 있다.
- Kile (<http://kile.sourceforge.net>)는 KDE용 \LaTeX 작업환경이다. 멋지고 완벽하다.
- Texmaker (<http://perso.club-internet.fr/pascal.brachet/texmaker/index.html>)는 여러 가지로 Kile와 비슷하지만 KDE가 없어도 된다.
- LyX (<http://www.lyx.org>)은 거의 WYSIWYG에 가까운 텍스트 처리기이다.^{*iv} 에디터로서는 그다지 뛰어나달 것이 없지만, 초보자에게는 의심할 바 없이 더 쉽게 느껴질 것이다.

^{*iii}이 가상의 워드 프로세서의 메뉴 항목 용어는 주로 한글과 MS 워드의 메뉴를 염두에 두고 번역하였다. 용어는 한글에서 취한 것도 있고 MS 워드에서 취한 것도 있다.

^{*iv}LyX의 한글 처리는 CJK-LyX을 이용하면 해결된다. 다만, CJK-LyX은 아직 Windows에서 사용하기는 어렵다.

- GNU T_EXMACS (<http://www.texmacs.org>)는 인상적인 UNIX용 L^AT_EX 워드 프로세서이다.
- WinEdt (<http://www.winedt.com>)는 아마 가장 인기있는 Windows 전용 L^AT_EX 작업환경일 것이다. 위의 자유 소프트웨어들과는 달리 이것은 shareware이다.
- Windows용 L^AT_EX 작업환경 프로그램의 자세한 리스트는 <http://home.arcor.de/itsfd/texwin.htm>에서 찾아볼 수 있다. 그 중에서도 TeXnicCenter와 WinShell을 추천한다.
- 마지막으로, 매킨토시 상에서의 L^AT_EX 운영에 관한 정보는 <http://www.esm.psu.edu/mac-tex/>을 참고하라.

위의 에디터들은 구문 하이라이팅은 물론이고 그외에도 L^AT_EX 원본 코드를 작성하는 데 유용한 여러 가지 기능을 갖추고 있다.

1.1.1 에디터가 지원하는 기능들

L^AT_EX은 조판기라는 사실을 분명히 해 두자. 잘라내기, 붙여넣기, 찾기, 바꾸기와 같은 기능은 에디터가 처리하도록 맡겨두고 있다. 표 1은, 특이한 것을 좋아하는 사람들에게 유명한 에디터들인 GNU emacs와 vim의 기본 키설정과 블랜드 IDE 키로 설정된 jed에서 동일한 기능을 각각 어떻게 구현하는지 요약한 것이다.

1.1.2 패키지 추가

다음은 teTeX이라는, 대부분의 GNU/Linux 배포판에 포함된 T_EX 시스템에 적용된다. MiK_TE_X에 대한 사항은 그 다음에 있다.

1. 패키지를 가까운 CTAN 미러 사이트에서 다운받는다. 대부분 gzip 압축 tar 아카이브(.tar.gz)이다.
2. L^AT_EX 패키지 위치를 찾는다. 대부분 \$TEXMF/tex/latex 일 것이다.
3. 이 디렉토리에서 위의 패키지의 압축을 푼다.
4. 만약 .sty 파일이 없다면, latex newstyle.ins 명령을 실행하여 .sty 파일을 만든다. 그런 다음 latex newstyle.dtx 명령을 실행하면 대개 관련 문서를 얻을 수 있다.
5. texhash 명령을 실행하여, teTeX이 새로운 패키지를 인식하도록 한다.

\$TEXMF의 정확한 위치는 시스템에 따라 달라진다. 대부분의 GNU/Linux 배포판에서는 /usr/share/texmf이다.

만약 root 권한이 없다면, 자신의 \$HOME 디렉토리의 하위 디렉토리에, 즉, \$HOME/texmf에, 패키지를 설치할 수 있다. .sty 파일을 그 곳에 저장하고 다음 내용을 \$HOME/.bash_profile에 추가하면 된다.

```
export TEXINPUTS="~/texmf:"
```

Action	Emacs	Vim	Jed
명령모드(command mode)	Alt-X	ESC	Alt-X
삽입모드(insert mode)	n/a	i a o O	n/a
줄편집모드(line editor mode)	n/a	:	n/a
파일관련			
파일 열기	Ctrl-X Ctrl-F	:e	Ctrl-KE
파일 끼워넣기	Ctrl-Xi	:r	Ctrl-KR
파일 저장	Ctrl-X Ctrl-S	:w	Ctrl-KD
새이름으로	Ctrl-X Ctrl-W name	:w name	Ctrl-KS
파일 닫기	Ctrl-XK	:q	Ctrl-KQ
change buffer	Ctrl-XB	bN	Ctrl-KN
취소	Ctrl-XU	u	Ctrl-U
다시하기	Ctrl-^	Ctrl-R	Ctrl-G Ctrl-U
나가기	Ctrl-X Ctrl-C	:qa!	Ctrl-KX
커서 움직이기			
한 단어 왼쪽으로	Alt-B	b	Ctrl-A
한 단어 오른쪽으로	Alt-F	w	Ctrl-F
줄의 처음으로	Ctrl-A	O	Ctrl-QS
줄의 끝으로	Ctrl-E	\$	Ctrl-QD
한 페이지 위로	Alt-V	Ctrl-U	Ctrl-R
한 페이지 아래로	Ctrl-V	Ctrl-D	Ctrl-C
버퍼(열린 파일)의 처음으로	Alt-<	1G	Ctrl-QR
버퍼(열린 파일)의 끝으로	Alt->	G	Ctrl-QC
줄번호 찾아가기	Alt-G n.	n.G	Ctrl-QI
지우기			
왼쪽 글자 지우기	Ctrl-H	X	BS
오른쪽 글자 지우기	Ctrl-D	x	Alt-G
왼쪽 단어 지우기	Alt-DEL	db	Alt-BS
오른쪽 단어 지우기	Alt-D	dw	Ctrl-T
줄끝까지 지우기	Ctrl-K	d\$	Ctrl-QY
줄 삭제	Ctrl-A Ctrl-K	dd	Ctrl-Y
찾기와 바꾸기			
찾기	Ctrl-S text	/text	Ctrl-QS
바꾸기	Alt-%	:s/old/new/g	Ctrl-QA
선택영역			
선택영역 표시 시작	Ctrl-SPACE	v	Ctrl-KB
잘라내기	Ctrl-W	D	Ctrl-KY
복사하기	Alt-W	Y	Ctrl-KH
붙여넣기	Ctrl-Y	P	Ctrl-KC

표 1: Emacs, Vim, Jed(IDE mode)의 기능별 키 설정

MiKTeX에서 패키지를 추가하려면, C:\localtexmf\tex 디렉토리 아래 \latex\newpackage 디렉토리를 만들고 관련 파일을 그 곳에 둔다. 위에서 말한 방식으로 한 다음, MiKTeX의 옵션관리자(MiKTeX Options)를 실행하고 “Refresh now” 버튼을 누른다. 또는, initexmf -u 명령을 명령행에서 실행해도 된다. 성공이다!^{*v}

1.2 문서 작성의 황금률

얘기를 계속하기 전에 다음 사항을 상기하자. L^AT_EX 2_ε로 문서를 만들 때 명심해야 할 사항이다.

1. 문서의 구조화에 익숙해지도록 하라. 편, 장, 절로 나누어서 문서를 작성하는 방식을 염두에 두고 있어야 한다. 학술 논문을 쓰는 경우가 아니라도 이것을 지키도록 하자.
2. L^AT_EX은 형태 설정에 사용되는 매개변수들이 사용자를 질리게 하는 면이 분명히 있다. 지금 쓰고 있는 글의 모양에는 크게 신경쓰지 말고, 내용에 집중하라.

한번 이 규칙을 믿고 적용해 보라. 그러면 인쇄된 최종 출력물은 신비롭게도 전문가가 만든 것 같은 모양이 나온다. 다시 말하거니와, L^AT_EX 입문서를 공부해보기 바란다.

이 규칙들은 대부분의 문서에 적용된다. 그러나, 작성하는 문서가 애시당초 비구조적인 것(회람, 메모 등)이라면... 좋다, 이 글을 계속 읽어보도록 하자.



제 2 절 파일(File) 메뉴

이 메뉴에 들어 있는 몇 가지 항목은 L^AT_EX과는 아무 관계가 없다. 파일 열기(File/Open), 저장하기(File/Save), 파일 닫기(File/Close) 같은 것은 에디터가 알아서 할 문제다.

2.1 새 문서(File/New)

새로운 빈 문서에 해당하는 L^AT_EX 소스는 다음과 같다.

```
\documentclass{article}
\begin{document}
% 여기에 본문이 온다.
\end{document}
```

L^AT_EX으로 작성된 문서는 본래부터 구조화된 문서이다. 다음은 보다 실용적인 예이다. 문서의 제목과 저자, 작성일자를 이용해서 타이틀(title)을 만들고, 개요와 목차를 설정하였다. 문서의 구조화 측면에서, 절의 제목을 써넣는 방법이나 인용(cite)과 참고문헌을 처리하는 것이 자동화되어 있다는 점에 주의하라.

^{*v}최근의 MiKTeX 배포판은 새로운 패키지의 설치가 훨씬 쉬워졌다. MiKTeX 2.2 이후 버전에서는 패키지 관리자 프로그램(MiKTeX Package Manager)이 별도로 제공된다. 인터넷에 연결되어 있거나 로컬 패키지 저장소가 있다면 패키지의 설치 삭제가 손쉽게 이루어진다.


```

\documentclass[a4paper,12pt]{article}
\begin{document}
\title{내 문서}
\author{홍~길~동}
\date{서울, \today}
\maketitle
\begin{abstract}
짧은 글을 하나 써 본다.
\end{abstract}
\tableofcontents
\listoftables
\listoffigures
\section{첫 번째 절}
\label{sec:start}
여기에 절의 본문이 온다. 자세한 사항은 \cite{Gonzato} 참조.
\section{End}
\label{sec:end} 여기가 문서의 끝이다. 제~\ref{sec:start}~절로
돌아가서 다시 읽어보시오.
\begin{thebibliography}{99}
\bibitem{Gonzato} Gonzato G. \textit{\LaTeX{} for Word Processor
Users}. CTAN, 2001.
\end{thebibliography}
\end{document}

```

부록 A에 더 많은 문서 서식(templates)의 예를 보였다.

2.2 새 이름으로(File/Save As...)

워드 프로세서의 Save As... 항목은 보통 다른 이름이나 다른 파일 포맷으로 변환하여 저장하는데 사용된다. \LaTeX 으로 작성된 문서를 다른 포맷으로 변환하고자 할 때 사용할 수 있는 도구로는 다음과 같은 것이 있다.

- latex2html: <http://saft sack.fs.uni-bayreuth.de/~latex2ht/>,
CTAN://support/latex2html
- latex2rtf: CTAN://support/latex2rtf

PDF 파일 만들기에 대해서는 제 2.7 절을 보라.

2.3 서식파일로 저장(File/Save As Template)

\LaTeX “서식(templates)”을 저장한다는 말이, 만약 새로운 \LaTeX 패키지 또는 클래스를 설계한다는 말이라면 이것은 쉽지 않은 일이 된다. 이 지침서의 범위를 넘어서는 것이다.

2.4 다른 문서 가져오기(File/Import)

다음 도구들은 다른 포맷의 문서를 \LaTeX 으로 변환하는 데 쓰인다.

- rtf2latex: [CTAN://support/rtf2latex](http://support.rtf2latex)
- html2latex: [CTAN://support/html2latex](http://support.html2latex)
- wvware는 MS Word 파일을 \LaTeX 을 비롯한 여러 가지 포맷으로 변환해주는 프로그램 모음이다. <http://wvware.sourceforge.net>.
- Abiword라는 프리소프트웨어 워드 프로세서는 MS Word 파일을 불러들여서 \LaTeX 으로 저장할 수 있다. <http://www.abisource.com>
- txt2tex : [CTAN://support/txt2tex](http://support.txt2tex)은 플레인 텍스트 파일을 \LaTeX 으로 변환해주는 꽤 괜찮은 도구이다.

그밖의 *2latex 변환기들을 같은 주소에서 찾아볼 수 있다.

2.5 편집용지 설정(File/Page Setup)

페이지 크기, 방향, 여백을 조절하는 일반적인 방법은 `\documentclass` 명령의 옵션 인자^{*vi}를 지정하는 것이다. 페이지 크기 옵션 인자는 a4paper, a5paper, b5paper, letterpaper, legalpaper, executivepaper를 쓸 수 있다. 용지 방향은 portrait(길게)가 기본값이고, landscape 옵션을 써서 landscape(넓게)로 바꿀 수 있다. 예를 들면 다음과 같이 한다.

```
\documentclass[a5paper,landscape,12pt]{article}
```

문서 전체의 여백은 다음 명령을 이용하여 설정한다.

```
\setlength{\leftmargin}{2cm}
\setlength{\rightmargin}{2cm}
\setlength{\oddsidemargin}{2cm}
\setlength{\evensidemargin}{2cm}
\setlength{\topmargin}{-1cm}
\setlength{\textwidth}{18cm}
\setlength{\textheight}{25cm}
```

geometry 패키지는 종이 크기, 여백 너비 등을 자유롭게 조절할 수 있게 해준다. geometry의 옵션은 너무 많아서 한꺼번에 나열하기 어려우므로, 패키지 문서를 읽어보기 바란다. 이 패키지를 이용하는 완전한 예를 아래에 하나 보였다. 이 보기에서 사용된 매개변수들은 다른 것과 상충하는 것도 있지만, 사용 방법을 보이기 위해서 예를 들어본 것이다.

```
\geometry{paperwidth=25cm}
\geometry{paperheight=35cm}
% or: \geometry{papersize={25cm,35cm}}
\geometry{width=20cm} % total width
\geometry{height=30cm} % total height
```

^{*vi}“옵션 인자”라는 것은 명령 뒤에 각진괄호로 [] 둘러싸서 넘겨주는 매개변수를 말한다. 반면, 중괄호 { } 안에 오는 것은 “매개변수” 또는 “인수”라고 번역하였다.

```
% or: \geometry{total={20cm,30cm}}
\geometry{textwidth=18cm} % width - marginpar
\geometry{textheight=25cm} % height - header - footer
% or: \geometry{body={18cm,25cm}}
\geometry{left=3cm} % left margin
\geometry{right=1.5cm} % right margin
% or: \geometry{hmargin={3cm,2cm}}
\geometry{top=2cm} % top margin
\geometry{bottom=3cm} % bottom margin
% or: \geometry{vmargin={2cm,3cm}}
\geometry{marginparwidth=2cm}
\geometry{head=1cm} % header space
```

옵션 인자로 다음과 같이 설정하는 방법도 있다.

```
\usepackage[left=3cm, right=2cm]{geometry}
```

2.5.1 머리글/바닥글(Page Setup/Headers and Footers)

fancyhdr 패키지는 `\pagestyle{fancy}`라는 새로운 명령을 제공한다. 이것은 현재 질(book.cls에서는 장)과 소절을 머리글에 기록하고 페이지 번호를 바닥글에 찍어준다. 제법 팬시(fancy)하다. 머리글과 바닥글은 사용자가 정할 수 있다. 세 부분으로 이루어져 있는데, 왼쪽 정렬된 부분, 가운데 정렬된 부분, 오른쪽 정렬된 부분이 그것이다. 이것을 설정하려면 다음 보기와 같은 방식의 명령을 사용한다.

```
\lhead{} % empty
\chead{Hello, world!}
\rhead{Page \thepage} % page number
\lfoot{}
\cfoot{\textbf{Hello!}}
\rfoot{}
```

2.6 프린터 설정(File/Printer Setup)

이 기능은 운영체제에 거의 전적으로 의존한다. 따라서 \LaTeX 이 다루는 영역이 아니다. 만약 UNIX 종류의 운영체제를 사용하고 있다면 다음과 같은 팁이 도움이 될 것이다.

- `lpr -P printername` 지정된 프린터로 인쇄.
- `lpr -# 10 10장` 인쇄.
- `lpr -r` 출력한 후 파일 삭제.

다른 팁은 아래를 보라.

2.7 프린트 미리보기(File/Print Preview)

TEX 입력 파일이 준비되었다면, 미리보기는 여러 가지 방법이 있다.

- .dvi 파일로 변환 (latex file.tex)한 다음 xdvi나 yap 같은 DVI 미리보기 프로그램으로 화면보기를 한다.
- .dvi 파일을 dvips를 이용하여 POSTSCRIPT로 변환한 다음 Ghostview나 그와 유사한 프로그램을 이용한다.
- dvipdfm을 이용하거나^{*vii} pdflatex를 직접 이용하여 Acrobat .pdf 파일을 만들어낸다.

내 견해로는 Acrobat .pdf 파일을 만드는 것이 가장 좋은 방법인 것 같다. .pdf는 폭넓은 이식성을 가지고 있다. dvipdf라는 스크립트(대부분의 Linux 배포판에는 포함되어 있다)는 .dvi를 .ps 파일로 바꾼 다음, 이를 .pdf로 변환해준다.^{*viii} pdflatex은 훨씬 흥미롭다. hyperref 패키지와 url 패키지를 사용하면 하이퍼링크를 포함하여 브라우징 가능한 .pdf 파일을 만들 수 있다. 제 4.15 절을 참고하라. 그러나 pdflatex을 사용하게 되면 몇몇 다른 패키지와 충돌을 일으킬 수 있는데, 더 자세한 것은 제 4.12 절에서 다루겠다.



2.8 인쇄(File/Print)

유닉스 시스템이라면 명령행에서 간단히 `lpr file.ps`라고 입력하면 된다. 또는 화면보기 프로그램의 메뉴를 열어서 ‘인쇄’를 선택한다.

2.9 버전 관리(File/Versions)

TEX 원본의 버전 관리를 위한 version 패키지가 있기는 하지만 RCS (Revision Control System)나 CVS (Concurrent Version Control System)과 같은 외부 유틸리티를 이용하거나 에디터에 통합되어 있는 기능을 이용하는 편이 낫다고 생각한다. CVS와 RCS에 대한 간략한 소개는 <http://www.linuxdoc.org/HOWTO/CVS-RCS-HOWTO.html>를 방문해 보라.

제 3 절 편집(Edit) 메뉴

이 메뉴는 TEX 자체의 기능보다는 에디터의 기능과 더 많은 관계를 가지고 있다. 우선 표 1에 요약된 내용을 보면, 흔히 쓰이는 몇 가지 에디터에서 Edit/Cut, Edit/Copy, Edit/Paste, Edit/Find, Edit/Replace 등이 어떤 키에 할당되어 있는지를 알 수 있다.

텍스트의 선택은 선택한 부분을 자르고 붙이기에도 쓰이지만 선택된 구역에 특정 스타일을 적용하기 위한 것일 때도 있다. TEX에서는 그 구역을 중괄호로 둘러싸거나 환경(environment) 안에

^{*vii} 한글이 포함된 PDF 파일을 만드는 가장 좋은 방법은 dvipdfm의 확장판인 DVIPDFMx를 사용하는 것이다. 조진환 님과 Hirata Shunsaku 님이 공동으로 개발하였던 dvipdfm-cjk가 발전한 프로그램이다. DVIPDFMx를 이용하면 한글 트루타입 폰트를 이용하여 검색 추출이 가능한 한글 .pdf 파일을 만들 수 있을 뿐 아니라, 그밖에 여러 가지 확장된 기능을 이용할 수 있다. <http://project.ktug.or.kr/dvipdfmx/>.

^{*viii} .ps를 .pdf로 변환하는 데는 Adobe Distiller나 PSTill과 같은 유틸리티를 쓰거나, Ghostscript의 ps2pdf 스크립트를 이용하는 방법 등이 있다. dvipdf는 먼저 dvips를 부른 다음, 위의 프로그램이나 스크립트를 이용하도록 하고 있는 스크립트이다. 반면, dvipdfm이나 DVIPDFMx는 .dvi를 .pdf로 번역해주는 역할을 한다.

두는 방식으로 한다. 예를 들면, 굵은 글씨체 속성을 텍스트 일부에 부여하기 위해서는 다음 가운데 한 가지 방법을 사용한다.

```
this is \textbf{bold text;}\\
this is also
{\bfseries bold text;}\\
\begin{bfseries}
this is bold text, too!
\end{bfseries}
한글로도 \textbf{굵은 글씨체}!
```

this is **bold text**;
 this is also **bold text**;
this is bold text, too! 한글로도 굵은 글씨체!

3.1 자동완성(Edit/Autotext)

자동완성기능이란, 예를 들면 “PS”라고 타이프했을 때 자동적으로 “POSTSCRIPT”가 입력되는 것을 가리킨다. 이것은 에디터의 기능이기는 하지만,^{*ix} L^AT_EX으로는 다음과 같이 할 수 있다.

```
\def\PS {\textsc{PostScript}}
```

이렇게 설정해 두면, \PS라고 입력된 모든 곳에서 \textsc{PostScript}로 바뀐다. 대소문자를 구별한다는 점에 주의하라.

제 4 절 삽입(Insert) 메뉴

4.1 쪽나누기, 문단나누기(Insert/Breaks)

- 줄바꿈 없는 빈 칸은 ~으로 표시한다.
- 개행(줄바꿈)은 \linebreak나 \newline, \\을 쓴다. \\ 다음에는 옵션으로 행간을 지정할 수 있다. 예를 들면 \\[1cm]와 같이. 더 자세한 것은 아래를 보라.
- 새로운 단락을 시작하려면 빈 줄을 하나 이상 넣든가, \par 명령을 쓴다.^{*x}
- 끝으로, 페이지를 나누려면 \newpage나 \clearpage를 쓴다.

\newline은 명령이 내려진 곳에서 새로운 행을 시작하지만, \linebreak 명령은 그 줄을 오른쪽 끝까지 전개한다.

줄 끝까지 벌어진다.\linebreak
 그렇지 않다.\newline
 아, 이제 알겠다.

줄 끝까지 벌어진다.
 그렇지 않다.
 아, 이제 알겠다.

\clearpage 명령은 \newpage와는 달리 아직 처리되지 않은 모든 떠다니는 개체(그림과 표)를 출력한 다음에 페이지를 나눈다. 한편, \cleardoublepage 명령은 \clearpage와 의미가 같으나 다음 쪽이 항상 홀수쪽에서 시작하도록 페이지를 나눈다. 떠다니는 개체에 대해서는 제 4.12 절을 보라.

^{*ix} 한글에서는 “상용구 기능”이라 한다.

^{*x}저자는 \\를 새로운 단락을 시작하는 명령으로 처리하고 있는데, 이 점은 역자와 의견이 다르다. 번역에서는 \\를 개행의 의미로 처리했다. 이 문제에 대해서는 역자주 *xxiii에서 더 자세히 말하겠다.

4.2 문단 번호(Insert/Enumerated List)

블릿문자나 숫자가 붙은 나열(문단머리 번호[기호] 붙이기)은 `itemize`와 `enumerate` 환경으로 표현한다. 나열 환경에서 문단 장식의 모양을 바꾸려면 `\item` 명령의 옵션인자로 지정하면 된다.

```
\begin{itemize}
\item[*] 별표 붙인 문단.
\item[-] 대시 붙인 문단.
\item[.] 도트 붙인 문단.
\end{itemize}
```

- * 별표 붙인 문단.
- 대시 붙인 문단.
- . 도트 붙인 문단.

또다른 방법은 제1수준에서 제4수준까지 나열형의 머리숫자² 모양을 재정의하는 것이다. 다양한 형식이 정의되어 있는데, `\arabic`이 보통 숫자에 해당하고, `\roman`은 로마 숫자 소문자(viii), `\Roman`은 로마 숫자 대문자, `\alph`와 `\Alph`는 각각 소문자, 대문자의 알파벳, 그리고 `\fnsymbol`이 있다. 마지막 `\fnsymbol`은 여기서 당장 설명하지 않겠다.

```
\begin{itemize}
\renewcommand{\labelitemi}{*}
\renewcommand{\labelitemii}{-}
\item 제1수준, 첫번째 항목
\item 제1수준, 두번째 항목
\begin{itemize}
\item 제2수준, 첫번째 항목
\item 제2수준, 두번째 항목
\end{itemize}
\item 제1수준, 세번째 항목
\end{itemize}
```

- * 제1수준, 첫번째 항목
- * 제1수준, 두번째 항목
 - 제2수준, 첫번째 항목
 - 제2수준, 두번째 항목
- * 제1수준, 세번째 항목

말하자면, 로마 숫자와 대문자 알파벳을 `enumerate` 나열 환경에서 쓰고 싶은 경우, 다음과 같이 하라:

```
\begin{enumerate}
\renewcommand{\labelenumi}{\Alph{enumi}}
\renewcommand{\labelenumii}{\roman{enumii}}
\item 제1수준, 첫번째 항목
\item 제1수준, 두번째 항목
\begin{enumerate}
\item 제2수준, 첫번째 항목
\item 제2수준, 두번째 항목
\end{enumerate}
\item 제1수준, 세번째 항목
\end{enumerate}
```

- A 제1수준, 첫번째 항목
- B 제1수준, 두번째 항목
 - i 제2수준, 첫번째 항목
 - ii 제2수준, 두번째 항목
- C 제1수준, 세번째 항목

`enumerate` 패키지를 이용하는 방법도 있다. 이 패키지는 `enumerate` 환경을 재정의하여, 옵션 인자를 지정할 수 있도록 하였다. 나열 목록의 머리숫자로 A a I i 1 문자를 쓸 수 있는데, 각각 `\Alph`,

²각각 `\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiv`에 해당한다.

문자	\LaTeX 명령
\$	<code>\\$</code> or <code>\textdollar</code>
&	<code>\&</code>
%	<code>\%</code>
-	<code>_</code> or <code>\textunderscore</code>
{	<code>\{</code> or <code>\textbraceleft</code>
}	<code>\}</code> or <code>\textbraceright</code>
<	<code>\$<\$</code> or <code>\textless</code>
>	<code>\$>\$</code> or <code>\textgreater</code>
\	<code>\textbackslash</code>
	<code>\textbar</code>
•	<code>\textbullet</code>
‡	<code>\textdaggerdbl</code>
†	<code>\textdagger</code>
¶	<code>\textparagraph</code>
§	<code>\textsection</code>
©	<code>\textcopyright</code>
^	<code>\textasciicircum</code>
~	<code>\textasciitilde</code>
~	<code>\$\sim\$</code>
®	<code>\textregistered</code>
™	<code>\texttrademark</code>
ª	<code>\textordfeminine</code>
º	<code>\textordmasculine</code>

표 2: 특수문자를 얻는 몇 가지 방법

`\alph`, `\Roman`, `\roman`, `\arabic`에 해당한다. 만약 옵션 인자에 텍스트를 쓰고자 할 때는 중괄호로 묶어주도록 한다.

```
\begin{enumerate}[{예제} I.]
\item 첫번째 항목 \label{item:first}
\item 두번째 항목
\item 세번째 항목. \ref{item:first}번 항목을
      볼 것.
\end{enumerate}
```

예제 I. 첫번째 항목
 예제 II. 두번째 항목
 예제 III. 세번째 항목. I번 항목을 볼 것.

4.3 특수문자(Insert/Special Character)

먼저, \LaTeX 에서 몇 가지 문자들은 특별한 의미를 가지고 있다는 것을 기억하자. 그러므로 그 글자를 식자하려면, `\`를 글자 앞에 써주거나, 수학 모드에서 쓰거나, 특별한 명령을 써야 한다. 표 2를 보라.

특별한 문자와 기호를 표시하는 방법은 여러 가지가 있다. 예를 들면 pifont 패키지는 `\ding`, `\dingfill`, `\dingline`, `\dinglist` 명령을 제공한다. 첫번째의 `\ding` 명령은 인수로 주어진 코드에 해당하는 Dingbat 문자를 찍어준다. 다른 명령들은 각각 `\fill`, `\line`, `\list` 명령을 주어진 Dingbat 문자를 사용하여 구현해주는 것이다.


```
\begin{dinglist}{43}
\item one
\item two
\item three
\end{dinglist}
```

☞ one
☞ two
☞ three

또는, 다음과 같이 하는 방법이 있다.

```
\begin{dingautolist}{172}
\item one
\item two
\item three
\end{dingautolist}
```

① one
② two
③ three

기호문자의 종류는 너무나 많아서 이 안내서에서 다 언급할 수가 없다. 다음 문서에 \LaTeX 에서 쓰이는 모든 기호문자를 제시해두고 있으므로, 참고하기 바란다. CTAN://info/symbols/comprehensive/

4.3.1 유로(EURO)화 기호

marvosym 패키지는 여러가지 재미있는 문자를 제공해주는데, 그 가운데는 유로화 단위인 €도 있다. `\EUR` 명령으로 얻을 수 있다.

4.4 수식(Insert/Formula)

\LaTeX 은 특히 수학식 처리에 강하다. 수학 기호를 본문에 삽입하려면 수식 부분을 `$`로 둘러싸면 된다.

```
I like math:  $x^n + y^n \neq z^n$ 
 $\forall n \neq 2$ 
is my favourite theorem.
```

I like math: $x^n + y^n \neq z^n \forall n \neq 2$ is my favourite theorem.

`displaymath`와 `equation` 환경은 본문과는 구분해서 수식을 별도로 식자한다. `equation` 환경은 상호참조를 위한 수식 번호를 덧붙여준다.*xi

```
페르마의 마지막 정리는
다음과 같다.
\begin{equation}
x^n + x^n \neq z^n \quad
\forall n \neq 2
\label{eq:fermat}
\end{equation}
식~\ref{eq:fermat}을
증명해보아라.
```

페르마의 마지막 정리는 다음과 같다.

$$x^n + x^n \neq z^n \quad \forall n \neq 2 \quad (1)$$

식 1을 증명해보아라.

*xi `equation`을 썼을 때, 일반적인 \LaTeX 문서(article, book 등)에서는 수식은 가운데 정렬되고 수식번호가 오른쪽에 붙는다. 수식번호를 왼쪽에 붙이려면 `\documentclass`의 옵션인자로 `[leqno]` 옵션을 추가한다. 그리고 `[fleqn]` 옵션은 수식을 왼쪽정렬(*flushleft-ed equation*)로 조판하도록 한다. 보다 복잡한 수식을 조판할 때는 `amsmath` 패키지가 쓰인다. 수식 조판에 관한 사항은 별도의 문서가 필요할 것이다.

4.5 각주(Insert/Footnote)

`\footnote[n]{footnote text}` 명령이면 충분하다. 옵션 인자(생략할 수 있는 인자) `[n]`은 임의의 각주 번호를 붙일 때 쓴다. 각주 번호 대신 기호나 특정 문자를 쓰려면 `\footnote` 명령에 해당하는 카운터를 재정의한다.

```
\renewcommand{\thefootnote}{read me!}
각주\footnote
{이게 각주다.}%
의 보기를 들어볼까.
```

각주^{read me!}의 보기를 들어볼까.

read me![!]이게 각주다.

이 방법을 쓰면 각주 번호를 로마 숫자나 보기좋은 기호문자로 바꿀 수 있다.

```
\renewcommand{\thefootnote}
{\Roman{footnote}}
여기\footnote{첫 번째}에
첫 번째 각주를 붙였고,
요기\footnote{두 번째}에
두 번째 각주를 붙였다.
\renewcommand{\thefootnote}
{\fnsymbol{footnote}}
끝.\footnote[8]{끝났다!}
```

여기^I에 첫 번째 각주를 붙였고, 요기^{II}에 두 번째 각주를 붙였다. 끝.^{††}

^I첫 번째
^{II}두 번째
^{††}끝났다!

`\fnsymbol{footnote}`로 각주번호를 기호 문자로 바꾸는 경우, 1에서 9까지의 값에 각각 아홉 종류의 기호 문자가 할당되어 있다. 각주번호에 따라 해당하는 기호문자가 붙는다. 그 순서는 다음과 같다. * † ‡ § ¶ || ** †† ‡‡.

같은 각주 번호를 두 곳 이상에 붙이고 하나의 각주로 참조하게 하려면 번호를 직접 써넣지 말고 `\footnotemark`를 이용해서 다음과 같이 하라.

```
이것\footnote{the first.}%
\newcounter{myfootnote}%
\setcounter{myfootnote}
{\value{footnote}}%
과 이것\footnote{the second.}은 각주이다.
첫 번째\footnotemark[\value{myfootnote}]
각주를 다시 읽어보자.
```

이것¹과 이것²은 각주이다. 첫 번째¹ 각주를 다시 읽어보자.

¹the first.
²the second.

주의: minipage는 mpfootnote와 thempfootnote라는 자체 카운터를 갖는다.

4.5.1 미주

endnotes 패키지는 모든 각주를 문서의 끝에 두게 해준다. 이것을 ‘미주’라고 한다. 전처리부에 다음과 같이 써넣는다.

```
\let\footnote=\endnote
```

그런 다음, 문서의 끝에 다음과 같이 지정한다.

```

\newpage
\beginngroup
\parindent 0pt
\parskip 2ex
\def\enotesize{\normalsize}
\theendnotes
\endgroup

```

그밖의 명령도 사용가능하다. endnotes.sty를 읽어보라.

4.6 목차 넣기(Insert/Indices)

목차, 표목차, 그림목차를 만들고 넣는 것은 \LaTeX 에서는 싱거운 일이다. 그저 다음 몇 줄을 처음 나오는 `\section` 또는 `\chapter` 명령 이전에 써주면 그만이다. 나머지는 \LaTeX 이 알아서한다.

```

\tableofcontents
\listoffigures
\listoftables

```

찾아보기를 만들려면 ^{*xii} makeindex와 makeidx 패키지를 이용한다. 같은 이름의 외부명령이 있고 `\index` 명령을 이용해서 색인에 포함되어야 할 단어를 지정해두면 \LaTeX 이 스스로 색인을 만든다. 다음 예는 역자가 색인을 만드는 방법의 예를 든 것이다.

```

\usepackage{makeidx}
\makeindex
\newcommand{\wordindex}[1]{\index{#1}#1}
%......
\printindex

```

여기서 `\makeindex` 명령은 색인을 만들라는 것이고, `\printindex`는 색인을 인쇄하라는 명령이다. 색인을 만들려면 \LaTeX 컴파일 시에 makeindex 명령을 명령행에서 한 번 실행한 다음 latex을 다시 실행하여야 한다. 즉, latex → makeindex → latex 순으로 컴파일하여야 한다. 위의 예에서 새로 정의된 `\wordindex` 명령은 예를 들면 `\wordindex{명령}`이라는 형태로 쓸 수 있다. 이제 “명령”이라는 단어가 색인에 추가된다.

4.7 수평 수직 간격(Insert/Vertical and Horizontal Space)

이 기능이 있는 워드 프로세서가 실제로 있는지 모르겠다. 사실 이것은 \LaTeX 이 매우 우아한 방법으로 해내고 있는, 다른 워드 프로세서에서는 찾아보기 힘든 기능이다.

공간 채우기는 텍스트를 가로나 세로의 가운데 위치시키려 할 때 쓰인다. 워드 프로세서에서 이렇게 하기는 좀 어려워서 상당한 시행착오를 거쳐야 한다. `\null` 명령이나 `~`로 위치를 설정하고 그 다음에 `\vfill` 또는 `\hfill` 명령을 써서 다음 보기와 같이 해보자.

^{*xii}이 단락은 역자의 보충이다.

```
one \hfill two\\
\vfill
~ \hfill three \hfill ~\\
\vfill
four \hfill five
\null
```

one	two
three	
four	five

일반적으로 \LaTeX 은 임의로 빈 공간을 넣을 수 없도록 하고 있다. 스페이스를 몇 개 반복해서 입력해도 그 스페이스만큼 벌어져서 조판되지 않는다. 그러나 문서가 흉하게 보여도 상관없다면 나누어지지 않는 스페이스를 `~`로 삽입할 수 있다.

또, `\hspace` 명령을 다음 보기와 같이 쓸 수 있다.

```
This is a \hspace{2cm}
2-cm-wide hole.
```

This is a	2-cm-wide hole.
-----------	-----------------

4.8 탭(Insert/Tabs)

tabbing 환경은 TAB 키를 치는 것과 비슷한 결과를 얻도록 해준다. 다음 예를 보라.

```
\begin{tabbing}
% let's set the tab positions
~ \hspace{1cm} \= ~ \hspace{2cm} \=
~ \hspace{2.5cm} \= \kill % discard text
Zero \> One \> Two \> Three \\
Zero \> One \> \> Three \+ \\ % go right
Zero \> Two \> Three \- \\ % go left
Zero \> One \> Two \\
\pushtabs % save tab positions
new tab 1{\ldots} \= new tab 2 \\
new \> tab \\
\poptabs % restore tab positions
Zero \> One \> Two \> Three
\end{tabbing}
```

Zero	One	Two	Three
Zero	One		Three
	Zero	Two	Three
Zero	One	Two	
new tab 1...		new tab 2	
new		tab	
Zero	One	Two	Three

tabular 환경과 table 환경도 볼 것.

4.9 상호참조(Insert/Cross Reference)

`\label`, `\ref`, `\pageref` 명령만 가지고도 본문에 레이블을 붙이고 상호참조를 하기에 충분하다. 레이블은 보통 `prefix:suffix` 형태로 쓴다. 여기서 `prefix`는 장(chapter)에 `cha`, 수식(equation)에 `eq`, 그림(figure)에 `fig`, 절/소절(sections)에 `sec`, 그리고 표(table)에 `tab` 등을 주로 쓴다.

특정 페이지에 있는 절, 표, 그림 등의 번호를 참조하려면 먼저 `\label`을 붙이고, `\ref` 명령을 이용한다. `\pageref` 명령은 그 그림 등이 있는 페이지번호를 참조할 수 있다.

```
\paragraph{Example.}
\label{par:example}
이 문단은
제~\pageref{par:example}~페이지의
제~\ref{par:example}~절에 나온다.
```

Example. 이 문단은 제 17 페이지의 제 4.9 절에 나온다.

label 이름의 구분자(prefix)는 임의로 붙여도 된다. 예컨대, 번호붙은 리스트에서 다음과 같이 해 볼 수 있을 것이다.

```
\begin{enumerate}
\item 첫 단계. \ref{item:end}번으로
갈 것. \label{item:start}
\item 다음 단계.(참조되지 않음)
\item 마지막 단계. \ref{item:start}번으로
돌아갈 것. \label{item:end}
\end{enumerate}
```

1. 첫 단계. 3번으로 갈 것.
2. 다음 단계.(참조되지 않음)
3. 마지막 단계. 1번으로 돌아갈 것.

4.10 난외주(Insert/Margin Notes)

굉장히 쉽다. \marginpar{text}를 쓴다.

4.11 글틀/글상자(Insert/Frame)

포스터를 편집하는 경우를 생각해 보자. 글과 그림을 페이지의 고정된 특정 위치에 놓을 방법이 있어야 할 것이다. textpos 패키지를 사용하면 그렇게 할 수 있다. 그림 7에 열거된 예제를 보라. (부록 A에 포스터를 편집하는 서식이 있다.)

좀더 쉬운 방법으로 minipage를 이용하는 방법이 있다. minipage란 이름 그대로 축소판 페이지를 말한다. 이 문단은 minipage 환경으로 둘러싸여 있는데, 더 정확하게 말하자면 boxedminipage 패키지가 제공하는 boxedminipage 환경으로 만들었다. 다음과 같이 한 것이다.

```
\begin{boxedminipage}[c]{0.6\linewidth}
... text ...
\end{boxedminipage}
```

4.12 그림삽입(Insert/Figure)

(\LaTeX 에서 그림 포함하기에 관한 참고 문헌은 epslatex.ps라고 알려져 있는 “Using Imported Graphics in $\text{\LaTeX}2_{\epsilon}$ ”이다.)

“그림(figure)”이라고 부르는 것은 실제 그림뿐 아니라 텍스트의 일부, 표 등등 figure 환경으로 둘러싸인 것을 모두 가리킨다. 예를 하나 들면 그림 1에 보인 바와 같다.

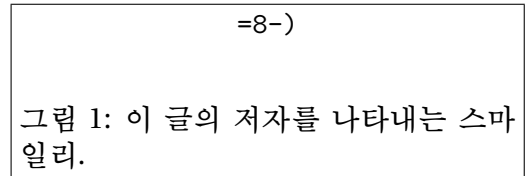
그림들은 원고에 지정된 정확한 그 위치에 나타난다는 것이 보장되지 않는다는 사실에 주의하라. 사실 워드 프로세서와의 주요 차이점이 그림이 고정된 위치에 있는 것으로 취급되지 않는다는 점이



```

\begin{figure}[htbp]
% [htbp] 위치지정자는 표가 놓일 위치를 지시하는
% 명령으로 각각 다음과 같은 뜻이다. here, top,
% bottom, separate page(별도페이지).
\begin{center}
\texttt{=8-)}
\end{center}
\caption{이 글의 저자를
    나타내는 스마일리.}
\label{fig:mysmiley}
\end{figure}

```



다. \LaTeX 에서 그림은 \LaTeX 자신이 스스로 결정하는 적절한 위치로 “떠다니게” 되어 있다. 그러므로, 본문에서는 “아래 그림”이나 “위의 그림”과 같이 참고하여서는 안되며, “그림 $\text{\ref{fig:label}}$ 을 볼 것” 이런 식으로 써야 한다.

이러한 속성 때문에 그림과 표는 떠다니는 개체 (*floats*)라고 불린다. 만약 이런 떠다니는 개체의 위치를 정확하게 특정하고 싶다면 `here` 패키지를 사용하여 위치 지정 옵션 인자를 `H`로 설정하면 된다.

그림이 Encapsulated POSTSCRIPT (.eps) 포맷이라면, `graphicx` 패키지를 써서 \LaTeX 원본 파일에 삽입할 수 있다. 그림 2에 보인 것과 같은 명령을 사용할 수 있다.

```

\begin{figure}
\begin{center}
\fbbox{\includegraphics
[width=0.5\textwidth, angle=-90]
{gnuplot.ps}}
\caption{A Gnuplot graph.}
\label{fig:gnuplot}
\end{center}
\end{figure}

```

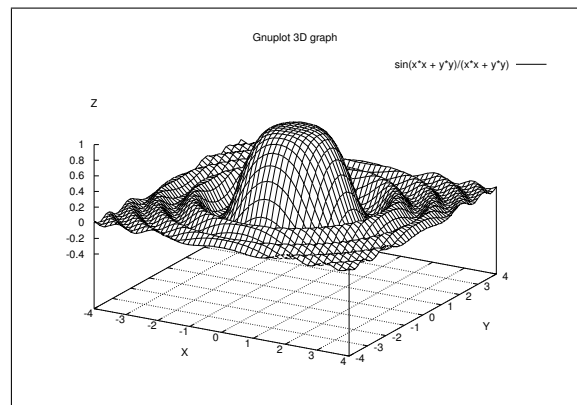


그림 2: A Gnuplot graph.

문서를 `latex`과 `dvips`로 처리하려 하는 경우, 그림은 EPS 파일만을 포함할 수 있다.^{*xiii} `pdflatex`은 PDF 포맷의 그림파일은 물론이고 JPG, PNG 그림도 처리한다.

.jpg, .gif, .png와 같은 일반적인 그래픽 포맷을 .eps로 변환시켜주는 프로그램이 많다. 그 중에 예를 들면 ImageMagick (<http://www.imagemagick.org>)과 GIMP (<http://www.gimp.org>)를 들 수 있다. 다만, 이 프로그램들이 만들어내는 POSTSCRIPT 파일은 크기가 엄청나다.

비트맵 그림을 작은 크기의 POSTSCRIPT 파일로 변환해주는 방법으로는 `jpeg2ps` (<http://www.pdf_lib.com/jpeg2ps/index.html>)나 `bmeps` (CTAN://support/bmeps)가 좋다. 앞의 것은 .jpg 파

^{*xiii} 반드시 그렇지는 않다. 외부 그림을 불러와서 포함하는 것은 .dvi 처리기의 기능이다. 그러므로 .dvi 처리기가 처리할 수 있다면 .jpg나 .gif를 포함하는 것도 가능하다. 다만, .eps가 가장 보편적으로 쓰이고 거의 모든 .dvi 처리기에서 사용할 수 있는 것일 뿐이다.

이 문서를 만든 DVIPDFMx를 사용하는 경우라면 EPS를 포함하는 데는 두 가지 방법이 있다. 하나는 `graphicx`를 엮은 후에 EPS 그림을 준비하여 DVIPDFMx로 하여금 처리하도록 하는 방법이고, 다른 하나는 `graphicx` 패키지를 [dvipdfm] 옵션과 함께 쓰면서 미리 EPS 그림을 PDF 파일로 변환시켜두는 방법이다. 더 자세한 것은 KTUG의 “그림 처리 TIP” 문서를 참고하라. <http://www.doeun.pe.kr/uselatex/ref-files/gfaq/>

일을 변환할 때는 가장 좋은 방법이고, 나중의 것은 더 많은 포맷의 그림파일을 다룰 수 있다.



만약 PDF_{TEX}을 사용한다면 .eps 그림을 반드시 .pdf로 바꾸고 원본에서도 .pdf 그림을 부르도록 수정해주어야 한다. EPS를 .pdf로 변환하는 데는 epstopdf를 이용한다.

.pdf와 .ps를 같은 원본 파일에서 컴파일러에 따라 선택적으로 사용하도록 하려면 다음과 같이 하라.^{*xiv}

```
% define the variable \ifpdf
\newif\ifpdf
\ifx\pdfoutput\undefined
  \pdffalse
\else
  \pdfoutput=1
  \pdftrue
\fi
...
% include the right options
\ifpdf
  \usepackage[pdftex]{graphicx}
  \pdfcompresslevel=9
\else
  \usepackage{graphicx}
\fi
...
% include the right graphic file
\ifpdf
  \includegraphics{file.pdf}
\else
  \includegraphics{file.eps}
\fi
```



18개 이상의 figure(떠다니는 개체)들이 대기열에 들어가서 처리되지 않고 있으면 ‘Too many unprocessed floats’라는 \TeX 에러를 만나게 된다. 이럴 경우 가장 간단한 방법은 \clearpage 명령을 세 개 또는 네 개의 그림 뒤에 붙여서 floats들을 처리하고 페이지를 나누게 하는 것이다. 참고로 morefloats 패키지를 쓰면 대기열에 저장할 수 있는 floats의 개수가 36개로 늘어난다.

4.12.1 그림과 표를 텍스트로 둘러싸기

잡지에서처럼 그림 주변으로 텍스트가 흐르게 하려면, wrapfig이나 picins, floatflt 등의 패키지를 이용한다.^{*xv}

^{*xiv}이 문서에서는 \newif\ifpdf를 정의하는 방법을 사용하였는데, 간단히 \usepackage{ifpdf} 하면, \ifpdf ... \else ... \fi를 바로 사용할 수 있다.

^{*xv}한글이 포함된 문서에서 wrapfig 패키지가 이따금 바라는 결과를 얻지 못하는 수가 있다. 이 경우에는 picins 패키지를 사용해볼 것을 권장한다. 이 문서의 원본은 wrapfig를 예로 들고 있지만 번역에서는 picins를 이용하는 방식으로 바꾸었다.

If you meet this guy, give him some money.

```
\newcommand\xsmile{\Huge\texttt{=8-)}}
\parpic{\xsmile}
{The reason may not be apparent to you,
but I can assure that your money
will end up in good hands.
I say again, if you meet this guy,
give him some money: he knows how to
use it properly. OK?}
```

If you meet this guy, give him some money.

=8-) The reason may not be apparent to you, but I can assure that your money will end up in good hands. I say again, if you meet this guy, give him some money: he knows how to use it properly. OK?

floatflt 패키지도 같은 역할을 하는 floatingtable 환경을 제공한다.

4.13 도형삽입(Insert/Shapes)

TeX은 \circle, \oval과 같은 명령을 사용해서 도형을 그릴 수 있게 하는 picture 환경을 제공한다. 내 생각에는 그래픽 환경이 아닌 텍스트 환경에서 그림을 그린다는 것은 너무나 어려운 일이라고 여겨지고, picture 환경도 여러 가지 한계를 가지고 있는 것으로 보인다. epic나 eepic 패키지를 이용하는 편이 더 낫고(이 패키지를 포함할 때는 순서를 지키도록 하라), 이 환경을 Xfig라는 그림그리기 프로그램(<http://www.xfig.org>)과 함께 사용하는 것이 좋다. 다만, Xfig는 UNIX환경에서만 사용가능하다.*^{xvi}

Xfig는 모양은 그다지 세련된 것이 아니지만 기능은 강력하다. 그 가운데 가장 좋은 점은 다양한 포맷으로 그림을 출력하는 것이 가능하다는 점인데, 그 포맷 중에는 eepic 매크로 형태도 포함된다. 또한 eepic 매크로를 매우 컴팩트한 POSTSCRIPT 파일로 변환할 수도 있다. 또 하나의 이점은 “special flag” 필드를 체크해두었다면 텍스트 개체를 TeX에게 처리하도록 할 수도 있다는 점이다. 이렇게 하면 그림에 TeX 수식과 기호문자를 써넣을 수도 있다.

그러니까, small.fig라는 이름의 그림을 하나 그렸다고 가정하자. File/Export...를 선택하고 Language 메뉴에서 “LaTeX picture + eepic macros”를 골랐다고 하자. 그러면 small.eepic라는 이름의 파일이 하나 더 생긴다. 이 그림을 문서에 포함하려면 다음과 같이 한다.

```
This is a picture
drawn with Xfig:\
\input{small.eepic}
```

This is a picture drawn with Xfig:



$$e^{i\pi} = -1$$

안타깝게도, eepic는 pdflatex과 함께 작동하지 않는다. 한 가지 대안은 그림을 “Combined PS/LaTeX (both parts)”로 내보내기(export)하는 것이다. 그러면 두 개의 파일을 얻는데 각각 small.pstex와 small.pstex_t이다. 두번째 파일을 small.tex으로 이름을 바꾼 다음에 파일의 시작 부분에 다음 몇 줄을 써넣는다.

^{xvi}Xfig는 Windows 환경에서도 사용할 수는 있지만, 윈도우용 X Server와 cygwin 환경과 같은 UNIX스타일의 환경을 구축해야 한다. Xfig와 유사한 java 프로그램으로 jfig가 있다. (<http://tech-www.informatik.uni-hamburg.de/applets/jfig/>)

```
\documentclass{article}
\usepackage{graphicx}
\pagestyle{empty}
\begin{document}
```

그리고 파일의 끝에 다음과 같이 써넣는다.

```
\end{document}
```

이제 이 파일을 컴파일한 다음 .ps 파일을 .eps로 변환한 후에 제 4.12 절에서 설명된 것과 같이 그림을 포함하면 된다.

만약 고난이도 그림을 그리고 싶다면 pstricks 패키지를 살펴보기 바란다. <http://www.tug.org/applications/PSTricks/>. 이 패키지는 \LaTeX 에서 POSTSCRIPT 그림을 직접 그리도록 해준다. 또 하나의 놀라운 프로그램인 ePiX (<http://mathcs.holycross.edu/~ahwang/current/ePiX.html>)는 \LaTeX 문서에 포함할 수 있는 과학용 플롯 그림을 작성해준다.

4.14 선그리기(Insert/Line)

`\rule` 명령은 길이와 두께를 지정하여 선을 그릴 수 있다.

```
This is a page-wide
rule:\\
\rule{\linewidth}{1pt}
but this one is shorter
and thicker:\\
\rule{2cm}{2mm}
```

This is a page-wide rule:

but this one is shorter and thicker:

“선” 중에서 흥미로운 것은 `\dotfill` 명령으로 얻을 수 있는 점선이다. 이것은 서로 관련된 것들을 연결할 때 흔히 쓰인다. 다음을 보라.

총액 `\dotfill` `\EUR~10`

총액 € 10

4.15 하이퍼링크 삽입(Insert/Hyperlink)

url 패키지는 URL을 기록할 수 있게 해주고, 적절하게 하이픈 처리도 할 수 있게 해준다. hyperref 패키지를 dvipdfm이나 pdf_latex과 함께 사용하면 url로 지시된 URL이 하이퍼링크 처리되어서 브라우저로 탐색할 수 있는 .pdf 파일을 만들 수 있다! 예를 들면, 이 문서는 다음과 같이 선언하여서 만든 것이다.

```
\usepackage[colorlinks,urlcolor=blue,filecolor=magenta]{hyperref}
\usepackage{url}
```

번역된 문서는 다음과 같이 설정하여 만들었다. 이 가운데 CJKbookmarks 옵션과, `\special` 코드는 DVIPDFMx에서 지원하는 기능이다.

```
\usepackage[dvipdfm,colorlinks,urlcolor=blue,filecolor=magenta,CJKbookmarks]{hyperref}
\AtBeginDvi{\special{pdf:tounicode KSCms-UHC-UCS2}}
\usepackage{url}
```

예를 들어보기로 하자.

```
\hypertarget{ctan}{CTAN}의 주 사이트는
\url{http://www.ctan.org}, 또는
\href{http://www.ctan.org}{CTAN://}이다.
```

```
\href{run:midifile.mid}{MIDI 음악 파일}을
들어보자.
```

```
\hyperlink{ctan}{여기}를 누르면 처음으로
돌아간다.
```

CTAN의 주사이트는 <http://www.ctan.org>, 또는 CTAN://이다.

MIDI음악파일을 들어보자.

여기를 누르면 처음으로 돌아간다.

`\hypertarget`과 `\hyperlink` 명령은 HTML 문서와 같은 내부 링크를 만들어준다. `\href` 명령은 URL이나 외부 파일의 링크에 쓰인다. `run:` 인자 사용에 주의하라. 이 하이퍼링크로 멀티미디어 플레이어, 오피스 응용프로그램 등등 어떤 외부프로그램이든 실행시킬 수 있다. 이 기능은 오직 Acrobat Reader에서만 작동하는 것으로 알고 있다.

리눅스나 UNIX 계통의 시스템에서는 Acrobat에게 외부 파일이 참조되었을 때 어떤 프로그램을 실행해야 하는지를 지시해두어야 한다. 예컨대 다음과 같은 설정을 `.mailcap`이나 `/etc/mailcap`에 넣어두도록 한다.

```
audio/midi;/usr/bin/timidity %s
audio/*; xmms %s
video/*; xine -pfhq %s
```

이밖의 다른 다양한 기능은 `hyperref` 패키지 문서를 읽어보기 바란다.

4.16 주석문(Insert/Comment)

TeX에서는 % 로 시작하는 줄 전체가 주석이 된다. 그외에도 `comment` 패키지를 이용하면 주석으로 처리할 수 있는 `comment` 환경을 쓸 수 있다.

제 5 절 모양(Format) 메뉴

일반적으로 문서의 쪽모양은 `\documentclass` 명령의 옵션 인자로 지정한다. 기본 글꼴 크기(10, 11, 12포인트), 종이 크기(a4paper, a5paper, b5paper, letterpaper, legalpaper, executivepaper) 종이 방향(portrait, landscape) 등등. 예를 들면 다음과 같다.

```
\documentclass[a5paper,landscape,12pt]{article}
```

추가로 별도의 글꼴 크기를 지정하려면 제 5.2.2 절에 설명된 것처럼 하면 된다.

5.1 행 간격(Format/Line Spacing)

setspace 패키지는 singlespace, onehalfspace, doublespace 환경을 제공한다. 그밖에 `\spacing{amount}` 명령 또는 환경은 지시된 amount 만큼의 값으로 행 간격을 설정해준다.*xvii

```
\begin{spacing}{2.5}
These two lines \\
are crazily spaced!
\end{spacing}
\begin{spacing}{1}
Much better, these lines\\
have a pretty space.
\end{spacing}
```

These two lines

are crazily spaced!

Much better, these lines
have a pretty space.

5.2 글자모양(Format/Character)

표 3에 표준 글자 모양을, 표 4에 글자 크기를 보였다.

Text attribute	Environment form	Example
<code>\textnormal</code>	textnormal	main document font
<code>\textrm</code>	rmfamily	roman
<code>\textit</code>	itshape	<i>italics</i>
<code>\emph</code>	n/a	<i>emphasis</i>
<code>\textmd</code>	mdseries	medium weight (default)
<code>\textbf</code>	bfseries	boldface
<code>\textup</code>	upshape	upright (default)
<code>\textsl</code>	slshape	<i>slanted</i>
<code>\textsf</code>	sffamily	sans serif
<code>\textsc</code>	scshape	SMALL CAPS
<code>\texttt</code>	ttfamily	typewriter
<code>\underline</code>	underline	<u>underline</u>
<code>\textsuperscript</code>	n/a	this is ^{superscript}
<code>\mathrm</code>	n/a	$x^n + y^n \neq z^n \forall n \neq 2$
<code>\mathbf</code>	n/a	$\mathbf{x^n + y^n \neq z^n \forall n \neq 2}$
<code>\mathsf</code>	n/a	$x^n + y^n \neq z^n \forall n \neq 2$
<code>\mathtt</code>	n/a	$x^n + y^n \neq z^n \forall n \neq 2$
<code>\mathit</code>	n/a	$x^n + y^n \neq z^n \forall n \neq 2$
<code>\mathnormal</code>	n/a	$x^n + y^n \neq z^n \forall n \neq 2$
<code>\mathcal</code>	n/a	$\S \backslash + \dagger \backslash \neq \ddagger \backslash \forall \backslash \neq \in$

표 3: 글꼴 속성

이탤릭체와 강조(emphasized)체의 차이점에 주목하라. 지금 이 문장은 이탤릭체로 쓰고 있는데, 이 단어는 강조 명령을 준 것이다. 여기서 보는 대로, `\emph` 명령은 직접적으로 이탤릭체를 쓰라

*xvii setspace 패키지를 로드하면 floats와 각주(footnote) 내의 행간은 모두 1.0으로 고정된다. 문서 전체의 행간을 지정하려면 setspace를 쓰지 않는 경우, L^AT_EX 명령인 `\linespread`를 이용할 수 있다. 한글 문서는 보통 영문보다 행간을 조금 더 띄어주어야 한다. `\linespread{1.5}`

는 타이포그래피 명령이 아니라 “논리적으로” 강조하는 명령이다.^{*xviii}

아래첨자는 수학적 모드에서만 쓰이는 것이 일반적이라는 사실을 알아두자. 보통 텍스트에서 아래 첨자를 쓰겠다면 다음과 같은 트릭이 있다.

```
this is
$_{\mbox{\footnotesize{subscript}}}$
```

this is subscript

Font size	Example
tiny	sample text
scriptsize	sample text
footnotesize	sample text
small	sample text
normalsize	sample text
large	sample text
Large	sample text
LARGE	sample text
huge	sample text
Huge	sample text

표 4: 글꼴 크기

5.2.1 밑줄 긋기

보통 밑줄은 쓰지 않는다. 밑줄은 옛날 타자기를 쓰던 시절의 유물이다. 그다지 보기에 좋지도 않다. 그래도 꼭 밑줄을 그어야 할 일이 있다면 ulem 패키지를 쓰면 된다. 밑줄 이외에도 몇 가지 재미있는 모양을 꾸밀 수 있다.^{*xix}

```
\uline{important}
\huline{중요함}
\uuline{urgent}
\huuline{긴급함}
\uwave{boat}
\huwave{보트}
\sout{wrong}
\hsout{잘못씀}
\xout{removed}
\hxout{지움}
```

important 중요함 urgent 긴급함 boat 보트
 wrong 잘못씀 ~~wrong~~ ~~잘못~~

주의사항: ulem 패키지는 \emph 명령을 재정의한다. 그러므로 \emph 명령이 쓰인 곳에 밑줄이 그려진다. 이것을 피하려면 다음과 같이 할 것.

```
\usepackage[normalem]{ulem}
```

^{*xviii} 즉, 영문에서, \emph 명령을 로마체에서 쓰면 이탤릭으로 식자되고, 이탤릭체에서 쓰면 로마체로 식자된다. 이 한글 번역 문서는 한글 기울임체 대신 순명조체로 식자하였다.

^{*xix} 아래 보기에서 한글 부분은 역자가 작성한 myulem을 이용한 것이다. ulem 패키지는 그대로는 한글²과 함께 쓸 수 없다. myulem 패키지에 관해서는 <http://faq.ktug.or.kr/mywiki/Karnes>를 참고하라.

5.2.2 글자 크기(Format/Character Size)

만약 표준 글자 크기로 충분하지 않다면 `extsizes` 패키지를 이용하면 편리하다. 이 패키지에서는 “확장된” 표준 문서 클래스 옵션을 제공하여, 8–12, 14, 17, 20포인트 문서를 작성할 수 있게 해준다. 예를 들면, 어떤 논문을 17포인트 폰트로 작성하고 싶다고 하자. 문서 전처리부분(preamble)에 다음과 같이 쓰면 된다.

```
\documentclass[17pt]{extarticle}
```

큰 폰트 크기를 얻는 다른 방법은 `type1cm` 패키지를 이용하는 것이다. 다음 보기는 사용법이다.

```
\fontsize{72pt}{72pt}\selectfont
No Smoking
```

(위의 보기는 이 페이지에 들어가기에는 너무 크다...)

`\fontsize` 명령의 두 인수 가운데 첫번째 것은 폰트 크기이고, 두번째 것은 기본행간격(baseline)이다. 또다른 방법으로 다음과 같은 것도 있다.

```
\resizebox{!}{1cm}{1-cm tall}
```

1-cm tall

5.2.3 글꼴 바꾸기(Format/Character Font)

\LaTeX 은 METAFONT 시스템에 의하여 필요한 글꼴이 생길 때마다 스스로 폰트를 만들어내는 방식으로 자체 폰트를 이용한다. 이렇게 함으로써 이식성을 보장하고 항상 좋은 출력물을 유지할 수 있다. 그러나, 우리는 이와는 다른 폰트에 익숙하다. 이를테면 Times체, Helvetica체, Sans Serif체 등등...

다행히도, \LaTeX 은 POSTSCRIPT 폰트도 사용할 수 있다. 다음 중에서 적당한 패키지를 골라서 이용하면 된다. `avant`, `avangar`, `bookman`, `chancery`, `charter`, `courier`, `helvet`, `helvetic`, `ncntrsbk`, `newcent`, `palatcm`, `palatino`, `pifont`, `times`, `utopia`, `zapfchan`. `\usepackage{times}`를 지정하고 결과를 얻어보라. 한 가지 문제라면 \LaTeX 이 수학식을 조판할 때는 Computer Modern 폰트를 쓰는 경우 가장 좋은 결과가 나온다는 점이다. POSTSCRIPT 폰트는 수학식에서 아주 조금 불만스러울지도 모르겠다.*^{xx}

위에 보인 패키지들은 문서 전체의 폰트를 바꾸어준다. POSTSCRIPT 폰트를 문서의 일부에만 적용하려면 글꼴 가족(font family)을 아래 예와 같이 지정해주어야 한다. 일반적인 글꼴 가족은 표 5에 정리해두었다.*^{xxi}

*^{xx}times와 palatino는 본문만을 POSTSCRIPT 글꼴로 바꾸기 때문에 수학식과 조화롭지 못하다고 말한 것이다. 최근의 PSNFSS에는 수식과 본문 서체에 통일성을 부여하는 좋은 패키지들이 많이 포함되었다. 예를 들면, times 계통의 mathptmx, txfonts, palatino 계통의 mathpazo, pxfonts, 그리고 utopia 계통의 fourier 등은 본문 글꼴과 어울리는 수학 서체를 식자해준다.

*^{xxi}한글 글꼴 가족은 표준화되어 있지 않다. 한글 \LaTeX 에서 사용할 수 있는 한글 글꼴 가족의 결합도 표 5에 보였다. 이 글꼴 가족들은 `\hfontfamily` 명령을 이용하여 얻을 수 있다.

가족	명칭
표준 글꼴 가족	
cmr	Computer Modern Roman
cmss	Computer Modern Sans Serif
cmtt	Computer Modern Typewriter
pag	Avantgarde
pbk	Bookman
phv	Helvetica
pnc	New Century Schoolbook
ppl	Palatino
ptm	Times
pcr	Courier
한글 \TeX 글꼴 가족	
mj	명조(바탕체)
gt	고딕(돋움체)
tz	타자(타자체)

표 5: 글꼴 가족



몇몇 폰트는 시스템에 따라 이용할 수 없을 수도 있다.

```
This is Computer Modern Roman,
{\fontfamily{phv}\selectfont
this is Helvetica!}
```

This is Computer Modern Roman, this is Helvetica!

또하나의 가능성은, 표준 \TeX 폰트를 POSTSCRIPT 폰트로 대체하는 것이다. 말하자면, Avantgarde 폰트를 Computer Modern Sans Serif 폰트를 써야할 곳에 대신 쓰고 싶은 경우가 있을 것인데, 이 경우 다음 명령들을 재설정하면 원하는 결과를 얻을 수 있다.^{*xxii}

- `\rmdefault` (roman)
- `\sfdefault` (sans serif)
- `\ttdefault` (typewriter)
- `\bfdefault` (boldface)
- `\mddefault` (medium)
- `\itdefault` (italics)
- `\sldefault` (slanted)
- `\scdefault` (small caps)
- `\updefault` (upright)

^{*xxii}한글 \TeX 에서는 `\mjdefault`, `\gtdefault` 등이 정의되어 있다.

다음 명령은 Sans Serif 글꼴을 Avantgarde 폰트로 대체해준다.

```
% Avantgarde replaces sans serif
\renewcommand{\sfdefault}{pag}
```

5.2.4 글자 색(Format/Character Colour)

color 패키지와 적당한 명령을 쓰면 단어에 색깔을 입힐 수 있다. 색깔의 기본값은 black, white, red, green, blue, cyan, magenta, yellow이다. 색깔을 따로 설정할 수도 있다.

```
\textcolor{red}{This is red.}\\
\color{blue}
This text is blue!\\
So is this. Let's change.\\
\definecolor{mygreen}
{rgb}{0.1,1,0.1}
\color{mygreen}
This is my shade of green!\\
\color{black}
\colorbox{cyan}{A cyan box}\\
\fcolorbox{blue}{green}
{A green box in a blue frame}
```

This is red.
 This text is blue!
 So is this. Let's change.
 This is my shade of green!
 A cyan box
 A green box in a blue frame

또, \pagecolor라는 명령도 있다... 어떤 결과를 얻을 것 같은가?

5.3 문단모양(Format/Paragraph)

문단이라는 것이 \LaTeX 에서는 무슨 뜻인지를 상기하자. 그것은 $\backslash\backslash$ 로 끝나거나 빈 줄이 뒤따르는 텍스트의 일부를 가리킨다.*xxiii

환경(environments)이라고 불리는 것은 \LaTeX 에서 텍스트 일부에 대하여 텍스트 정렬이나 글꼴 선택 등의 속성을 부여하는 방법이다. 워드 프로세서에서 텍스트 일부를 마우스로 선택한 다음, 메뉴에서 부르거나 버튼을 눌러서 텍스트 속성을 지정하는 것과 같은 것으로 보면 된다. 비슷한 일을 하기 위해 중괄호로 텍스트를 둘러싸는 방법도 있다.

환경은 일반적으로 다음 형식으로 쓴다.

```
\begin{environment}
...text goes here...
\end{environment}
```

어떤 문단을 가운데 정렬하고 싶다면 center 환경을 사용한다.

```
\begin{center}
가운데 정렬된 텍스트
\end{center}
```

가운데 정렬된 텍스트

*xxiii $\backslash\backslash$ 로 끝나는 텍스트는 엄밀한 의미에서 문단이라고 할 수 없다. 문단은 $\backslash\text{par}$ 로 끝나거나 빈 줄이 뒤따르는 텍스트의 일부라고 해야 한다. $\backslash\backslash$ 는 강제개행의 의미라고 보아야 옳을 것이다. 문단 뒤에는 개행이 오지만, 개행이 문단 구분은 아니다.

Environment	Purpose
array	배열(수학)
center	가운데 정렬
description	설명 형식의 리스트
enumerate	숫자 붙은 리스트
eqnarray	정렬된 수식
equation	수식 표현
figure	떠다니는 그림
flushleft	왼쪽 정렬
flushright	오른쪽 정렬
itemize	기호 붙은 리스트
letter	편지쓰기
list	일반적인 리스트 환경
minipage	축소판 페이지
picture	텍스트, 화살표, 선, 원 등을 포함한 그림
quotation	단락나누기를 포함한 인용문 환경
quote	단락나누기가 없는 인용문 환경
tabbing	강제 맞춤(탭)
table	떠다니는 표
tabular	표그리기
thebibliography	참고문헌 목록
theorem	정리, 보조정리 등
titlepage	직접 작성한 타이틀 페이지
verbatim	타이프한 대로 보이기
verse	시나 운문 등을 쓰는 환경

표 6: 표준 \LaTeX 환경들.

표준적인 환경은 표 6에 정리되어 있다. 이어지는 절에서 이 환경 가운데 어떤 것을 언제 쓰는지 보여주겠다.

5.3.1 문단의 가로 정렬(Format/Paragraph Horizontal Alignment)

문단 정렬의 기본값은 좌우정렬(justified)하는 것이다. 왼쪽정렬, 오른쪽정렬, 가운데정렬하려면 `flushleft`, `flushright`, `center` 환경을 사용한다. 여기에 대응하는 명령은 각각 `\raggedright`, `\raggedleft`, `\centering` 명령들이다. 환경과는 달리 명령형 선언은 새로운 문단을 시작하지 않는다.

5.3.2 문단의 간격 벌림(Format/Paragraph Vertical Alignment)

\LaTeX 의 문단 구분방식은 워드 프로세서를 사용하던 사람에게 혼란을 가져오는 경우가 많다. 워드 프로세서에서는 Enter 키를 치면 문단과 문단 사이에 한 줄의 빈 공간이 생기지만, \LaTeX 에서는 여러 줄을 벌려도 다만 문단만 나누어줄 뿐이고 문단 사이를 벌려주지는 않는다. 즉, 빈 줄과 여러 개의 공백은 한 개의 빈 줄이나 공백과 같이 취급된다. 문단 간의 간격을 벌리려면 `\smallskip`, `\medskip`, `\bigskip`과 같은 명령을 설정해야 한다.

이보다 더 넓게 문단 간격을 벌리려면, `\vskip{parameter}` 명령을 사용하라. 아래에 보기를 들었다.

```
These paragraphs will be
separated by 1.3 cm:\\
\vskip 1.3cm
there is a 1.3 cm gap above me.
```

These paragraphs will be separated by 1.3 cm:

there is a 1.3 cm gap above me.

`\vskip` 명령은 오직 문단 사이에만 효과가 있다는 사실을 명심하라. 즉, 새로운 페이지가 시작될 때는 `\vskip`이 효과가 없다. 새로운 페이지를 시작한 다음 예컨대 1.5 cm 정도를 페이지 상단에서 띄우고 싶다면 `\vskip` 명령을 쓰기 전에 띄우기 시작하는 위치를 `\null`로 “표시”해야 한다.

```
\null
\vskip 1.3 cm
This text comes after 1.3 cm...
```

This text comes after 1.3 cm...

마지막으로, `\vfill` 명령을 두 문단 사이에 쓰면 두번째 문단이 페이지의 바닥까지 내려가도록 두 문단 사이를 벌려준다. 다음 보기를 보라.

```
페이지 꼭대기에 오는 텍스트{\ldots}
\vfill
{\ldots}그리고 바닥에.
```

페이지 꼭대기에 오는 텍스트...

...그리고 바닥에.

5.3.3 문단 여백(Format/Paragraph Margins)

일반적으로 여백(margins)은 제 2.5절에서 보인 바와 같이, 전체 문서에 대해서 설정된다. 본문의 일부에 대해서 여백을 바꾸거나 재설정해도 효과가 발생하지 않을 것이다. 만약 어떤 문단에 여백을 새로 조절하려면 다음과 같은 새로운 환경을 정의해서 써야 한다.^{*xxiv}

```
\newenvironment{margins}[2]
{
  \begin{list}{} {
    \setlength{\leftmargin}{#1}
    \setlength{\rightmargin}{#2}
  } \item }
{\end{list}}
```

위와 같이 설정한 다음, 본문에서 새로 정의된 margins 환경을 쓸 수 있다.

^{*xxiv}chngpage 패키지를 이용하면 이 일을 보다 쉽게 할 수 있다.

As you can see, this paragraph
has normal margins.
`\begin{margins}{0.5cm}{1cm}`
But please note that this
paragraph has custom margins.
`\end{margins}`

As you can see, this paragraph has normal
margins.

But please note that this paragraph
has custom margins.

5.3.4 문단 들여쓰기(Format/Paragraph Indentation)

한 문단의 첫 줄 들여쓰기 크기를 설정하려면 `\parindent` 명령을 이용한다. 다음 보기는 들여쓰기 값을 1cm로 설정하는 것이다.

```
\setlength{\parindent}{1cm}
```

`\indent`, `\noindent` 명령은 그 다음에 나오는 문단의 들여쓰기를 허용하거나 해제한다. 끝으로, 문서 전체의 문단 사이 간격은 `\parskip`으로 설정한다.

```
\setlength{\parskip}{3pt}
```

5.4 문단 테두리와 음영(Format/Paragraph Border and Shade)

문단이나 일부 단어를 패선 테두리로 둘러싸려면 `framed` 패키지를 쓰거나 `\parbox` 명령을 사용한다. 이 때는 `calc` 패키지가 필요할 수도 있다.

아래는 `framed` 패키지를 이용한 가장 간단한 방법이다.

```
\setlength{\FrameRule}{2pt}
\setlength{\FrameSep}{5pt}
\begin{framed}
  문단에 테두리!
\end{framed}
\definecolor{shadecolor}{rgb}
{0.9,0.8,1}
\begin{shaded}
  문단에 음영,
  이것이 좋은가요?
\end{shaded}
```

문단에 테두리!

문단에 음영, 이것이 좋은가요?

마찬가지 결과를 얻기 위해서 `boxedminipage` 패키지가 제공하는 `boxedminipage` 환경을 이용할 수 있다. 좀더 자세히 설명하자면,

```
\framebox{
  \begin{minipage}[c]{\linewidth}
    테두리 쳐진 텍스트
  \end{minipage}
}
```

이 명령은 boxedminipage 환경과 완전히 동일한 역할을 한다.
그리고, \parbox를 이용한 예도 하나 들어두자.

```
\noindent
\fbbox{
  \parbox{\linewidth
    -2 \fbboxsep -2 \fbboxrule}
  {테두리쳐진 텍스트,
    하나 더!}
}
```

테두리쳐진 텍스트, 하나 더!

\linewidth 명령은 minipage의 폭(width)을 본문의 폭과 같게 만든다. 물론 특정의 폭(길이)를 설정할 수도 있다.

끝으로, 일부 텍스트의 길이를 얻어서 그 텍스트에 테두리를 치게 하려면 다음과 같이 한다.

```
한 단어에
\framebox[\width]{테두리를}
칠 수도 있다.
```

한 단어에 테두리를 칠 수도 있다.

옵션 인자를 수정함으로써 테두리 폭을 조절할 수 있다.

```
이번에는
\framebox[2\width][r]{테두리쳐진}
단어
```

이번에는 테두리쳐진 단어

두번째 옵션 인자는 테두리 속에 들어가는 부분의 정렬방향(이 보기에서는 오른쪽[r])을 지시하고 있다는 점을 주의해서 보라.

5.5 색상(Format/Colour)

테두리를 친 문단이 있다면 거기에 색깔을 입힐 수도 있다. 다음과 같이 하자.

```
\colorbox{yellow}{
  \begin{minipage}
    {0.8\linewidth}
    노란색의 미니페이지
  \end{minipage}
}
```

노란색의 미니페이지

색깔에 대해서는 제 5.2.4 절을 더 살펴보라.

5.6 다단편집(Format/Columns)

\twocolumn과 \onecolumn 명령은 이름 그대로 1단에서 2단으로, 2단에서 1단으로 단의 수를 바꾼다. 다만, 항상 새로운 페이지를 시작한다. 단 설정은 \documentclass의 옵션인자로 지정할 수도 있다. 이것으로 만족하지 못한다면 multicols 패키지를 사용해볼 수 있다. 이 패키지에서는 다단편집을 위한 multicols 환경을 제공한다. 다음과 같은 명령으로 문서를 2단으로 조판할 수 있다.

```

\columnseprule=1pt
\begin{multicols}{2}[\subsection{\entry{Format}{Columns}}]
The commands \cmd{twocolumn} ...
\end{multicols}

```

단 간격은 `\columnsep` 명령으로 조절한다. 단 사이에 줄을 넣고 그 두께를 바꾸려면 `\columnseprule`을 설정하면 된다. 각진괄호의 옵션 인자로 주어지는 텍스트에 대해서는 단설정 환경이 영향을 미치지 않는다.^{*xxv}

제 6 절 표(Table) 메뉴

표 그리기는 좀 복잡한 문제인데... *table*은 (제 4.12 절에서 설명하였듯이) 한 페이지 안에 놓이는 떠다니는 개체이다. 이 환경은 대부분 tabular 환경으로 만들어지는 표가 온다. 물론 다른 것이 들어가도 상관없기는 하지만. 표의 가로폭은 그 내용의 폭에 맞추어서 설정되도록 되어 있다.

다음은 일반적인 표의 형식이다.

```

\begin{table}[htbp] % placement: here, top, bottom, separate page
% \begin{small}      % sets the table font
\begin{center}      % optional
% 4-column table; alignment is left, centered, right, fixed width
\begin{tabular}{|l|l|c|rp{4cm}|}
\hline                % horizontal line
\textbf{Left} & \textbf{Centre} & \textbf{Right} & \textbf{4 cm} \\
\hline
row 1, col 1 & row 1, col 2 & row 1, col 3 & row 1, col 4 \\
\cline{1-2}          % horizontal line spanning columns 1-2
row 2, col 1 & row 2, col 2 & row 2, col 3 & row 2, col 4 \\
\cline{1-2}
\multicolumn{2}{|c|}{spanning two columns} & row 3, col 3 &
row 3, col 4 \\
\cline{1-3}
row 4, col 1 & row 4, col 2 & row 4, col 3 & ~ \hfill right \\
% force a space with "\ "
row 5, col 1 & row 5, col 2 & row 5, col 3 & left \hfill ~ \\
row 5, col 1 & row 5, col 2 & row 5, col 3 &
~ \hfill centre \hfill ~ \\
\hline
\end{tabular}
%\caption{A sample table.}
\caption{표 예제}
% labels are used for cross references;
% for example, "see Table~\ref{tab:samplatab}"
\label{tab:samplatab}
\end{center}

```

^{*xxv} 다단편집은 조금 복잡한 문제가 몇 가지 있다. 2단 편집의 경우 마지막 페이지의 텍스트 밸런싱(두 단의 세로 길이를 맞추는 것)이라든지, 한 페이지에서 두 종류 이상의 다단 텍스트를 배열하는 것, 또 그림이나 수식을 다단에서 적절하게 배치하는 것 등이 그것이다. 이 글에서 언급한 multicols 패키지 이외에 flushend, cuted, balance 등의 패키지 사용법을 참고하기 바란다.

```
% \end{small}
\end{table}
```

표 7에서 그 결과를 볼 수 있다.

Left	Centre	Right	4 cm
row 1, col 1	row 1, col 2	row 1, col 3	row 1, col 4
row 2, col 1	row 2, col 2	row 2, col 3	row 2, col 4
spanning two columns		row 3, col 3	row 3, col 4
row 4, col 1	row 4, col 2	row 4, col 3	right
row 5, col 1	row 5, col 2	row 5, col 3	
row 5, col 1	row 5, col 2	row 5, col 3	left
row 5, col 1	row 5, col 2	row 5, col 3	centre

표 7: 표 예제

가끔 표의 가로폭이 너무 길어서 페이지에 넘칠 때가 있다. 그럴 경우에는 rotating 패키지가 제공하는 sidewaystable 환경을 써보자. rotating 패키지는 특정 셀의 내용을 주어진 각도로 회전시킬 수도 있다. 그리고, tabularx 패키지는 고정폭 표를 그릴 수 있도록 해준다. X 컬럼지시자는 필요에 따라 컬럼 폭을 늘릴 수 있도록 하는 것이다.

다음은 예제이다.

```
\begin{sidewaystable}
\begin{tabularx}{7.5cm}{|l|X|X|}
\hline
\textbf{normal} & \textbf{tilted} &
\textbf{wider} \\
\hline
normal & \rotatebox{30}{I'm tilted!} &
I'm wider \\
\hline
\end{tabularx}
\end{sidewaystable}
```

wider	wider
tilted	I'm tilted!
normal	normal

표준적인 tabular 환경은 한 페이지를 넘어가지 못한다. 이 제한을 극복하도록 하는 패키지가 여럿 있는데, longtable, supertabular, xtab 등을 시험해볼 수 있을 것이다.

표에 색깔을 입히려면 colortbl 패키지를 쓰면 된다.


```

Colour by row:\\vskip 2mm
\begin{tabular}{|l|c|r|}
\hline
\rowcolor{cyan}
one & two & three\\
\rowcolor{green}
one & two & three\\
\rowcolor{yellow}
one & two & three\\
\hline
\end{tabular}

```

Colour by row:

one	two	three
one	two	three
one	two	three

```

Colour by column:\\vskip 2mm
\begin{tabular}
{|>{\columncolor{cyan}}l|
>{\color{red}
\columncolor{green}}c|
>{\columncolor{yellow}}r|}
\hline
one & two & three\\
one & two & three\\
one & two & three\\
\hline
\end{tabular}

```

Colour by column:

one	two	three
one	two	three
one	two	three

6.1 행 사이에 임의의 간격 부여하기

행 높이는 텍스트의 높이(height)에 따라 스스로 조절된다. 어떤 행의 앞에 여분의 공간을 넣고 싶다면, 행이 시작되기 전에 길이가 0이고 임의의 높이를 가지는 `\rule`을 그려주는 트릭을 써볼 수 있다. ^{*xxvi} 행의 뒤에 여분의 공간을 넣고 싶다면 `\\`에 옵션인자로 임의의 공간을 지정해주면 된다.

```

\begin{tabular}{|l|l|l|}
one & two & three\\
0.3 centimeters & \textbf{after} & & \\
this line\\[0.3cm]
one & two & three\\
one & two & three\\
\rule{0pt}{1.2cm}1.2 centimeters & & \\
\textbf{before} & & this line\\
\end{tabular}

```

one	two	three
0.3 centimeters	after	this line
one	two	three
one	two	three
1.2 centimeters	before	this line

^{*xxvi} 위치를 확보하기 위해서 넣는 폭이 0인 선을 Strut라고 한다. 표(tabular)의 행과 행 사이 간격을 전체적으로 조정하려면 `\arraystretch`라는 상수를 재정의한다. `\renewcommand{\arraystretch}{1.6}`.

6.1.1 소수점 정렬

tabular 환경에서 특별한 것 하나는 숫자들을 소수점을 기준으로 정렬하는 문제이다.

가장 간단한 방법은 @ 컬럼지시자를 쓰는 것으로서, 실제로 표가 숫자만 포함하고 있을 때는 유용하다. & 컬럼분리자를 소수점으로 바꾸어놓는다.

```
\begin{tabular}{r@{.}l}
3&14159\\
1&61803\\
1&41421\\
100&00000
\end{tabular}
```

3.14159
1.61803
1.41421
100.00000

또다른 방법은 dcolumn 패키지를 이용하는 것이다. dcolumn 패키지를 이용하면 D 컬럼 지정자를 사용할 수 있다. D 지정자는 세 가지 인수가 있는데, 하나는 L^AT_EX 원본과 출력에서 쓰이는 소수점 분리자(대부분, ‘.’이다)를 설정하는 것이고, 다른 하나는 소수점 아래 자릿수를 지정하는 것이다. 세번째 것은 지정하지 않아도 되는 것으로, 소수점 왼쪽과 오른쪽 자릿수를 설정할 수 있게 한다. 그리고 세번째 인수를 -1로 하면, 컬럼 내용이 소수점 분리자를 중심으로 중앙정렬된다.

이 지정자를 쓰면 자동으로 수식 모드에 있는 것으로 간주되므로, 일반 텍스트를 써넣으려면 \mbox 안에 두어야 한다.

```
\begin{tabular}{|D{.}{,}{4.2}|}%
D{.}{.}{5}|D{.}{.}{-1}|}
\hline
\mbox{One} & \mbox{Two} &
\mbox{Three} \\
10.33 & 10.33 & 10.33 \\
1000 & 1000 & 1000 \\
5.1 & 5.1 & 5.1 \\
3.14 & 3.14159 & 3.14159 \\
\hline
\end{tabular}
```

One	Two	Three
10,33	10.33	10.33
1000	1000	1000
5,1	5.1	5.1
3,14	3.14159	3.14159

6.2 slashbox로 셀에 선 긋기

이 패키지를 사용하면 \backslashslashbox와 같은 명령을 쓸 수 있다.

```
\begin{tabular}{|l|l|l|}
\hline
\backslashslashbox[2cm]{Lesson}{Date} &
Monday & Tuesday \\
\hline
Stratigraphy & room A & room A \\
Chemistry & room B & Lab $\alpha$ \\
Physics & room C & Lab $\beta$ \\
\hline
\end{tabular}
```

Date	Monday	Tuesday
Lesson	room A	room A
Stratigraphy	room B	Lab α
Chemistry	room C	Lab β
Physics		

6.3 \LaTeX 표로 데이터 가져오기

데이터 파일은 늘상 다루는 일용할 양식이나 다름없다. 대부분의 데이터 파일은 여러 컬럼으로 숫자가 나열된 단순한 ASCII 텍스트이거나, 어떤 것은 스프레드시트 파일이거나 하다. 거의 모든 스프레드시트 프로그램은 시트를 ASCII 형식의 .cvf 포맷으로 출력할 수 있다. 이 파일에서는 데이터의 값들이 ‘ ’ 문자로 분리되어 있다.

데이터 파일을 \LaTeX 표로 변환하는 일은 정말 지루한 작업이다. 아래 소개하는 UNIX 스크립트는 데이터 파일을 임의의 갯수의 컬럼을 가진 표로 변환시켜준다.^{*xxvii} .cvf 파일에도 적용된다.

```
#!/bin/sh

# dat2tex: converts tabular data to a tabular environment

if [ $# != 1 ]; then
    echo "Usage: $0 <datafile>"
    exit 1
fi

# is this a cvs file?
grep ";" $1 > /dev/null
if [ $? = 0 ]; then
    AWK="awk -F;"
else
    AWK=awk
fi

# ok awk, make my day
$AWK '{if (1 == FNR) { \
    printf "\\begin{tabular}\\{\\{"; \
    for (i = 1; i <= NF; i++) {printf "1"; \
    printf "\\}\\n"
    }
    for (i = 1; i < NF; i++) \
    {printf $i" & "} printf $NF" \\\n"} \
    END {printf "\\end{tabular}\\{\\n"}' $1

# end of dat2tex
```

제 7 절 도구(Tools) 메뉴

7.1 대량 메일 작성(Tools/Mail Merges)

유용하고 시간을 절약해주는 메일 머지 도구를 \LaTeX 에서 구현하려면 textmerg 패키지를 이용한다. 간단한 문서가 하나 있다고 하자. 이 문서를 내용은 그대로 두고 성과 이름, 호칭만 다르게 하여 여러 사람에게 보내려 하는 경우이다.

^{*xxvii} 김주명 님의 Excel 매크로는 Excel에서 작성된 표를 Tabular 코드로 바꾸어주는 유용한 프로그램이다. <http://faq.ktug.or.kr/mywiki/Excel2Tabular>를 보라. 홍석호 님의 Math 'N Table도 유용하다. http://physics.khu.ac.kr/~reds/Hpack_Project/k-download.htm.

\Name, \Surname, \Title이라는 세 개의 필드(*fields*)를 먼저 정의해야 한다. 텍스트에서 달라지는 부분이다. 여기에 해당하는 값들은 data.dat라는 외부 파일에서 가져온다.

```
\documentclass{article}
\usepackage{textmerg}
\begin{document}
% let's declare the variable fields:
% \Void is for empty lines
\Fields{\Name\Surname\Title-\Void}
\Merge{data.dat}{%
Dear \Title{} \Surname,\\
may I call you \Name?\\
Yours,\\
\hspace{3cm}Guido\clearpage}
\end{document}
```

네번째 필드인 \Void는 실제로는 필요없는 것이지만 동작 지정을 위한 인수로 마련되어 있다. 이 값 앞에 마이너스 부호가 붙어 있으면, 데이터 파일에서 그 부분이 비어 있음을 뜻한다. 레코드 값을 구분하는 데는 간단히 빈 줄을 넣어주면 된다.

data.dat 파일의 내용은 다음과 같다.

```
Guido
Gonzato
Dr.

Francesco
Mulargia
Prof.

Carol
John-Paul II
Pope
```

다 됐다. 각각의 수신인별로 한 페이지씩 텍스트를 출력해준다.

7.2 레이블 만들기(Tools/Labels)

메일 머지가 어렵지 않았다면, 레이블 만들기는 아주 간단하다. 3×8 접착 레이블 용지에 20개의 똑같은 레이블을 만들려 한다면, 짐작했겠지만 labels 패키지를 사용하면 된다. 다음 보기에서 간단히 10개의 간단한 레이블과 10개의 테두리친 레이블을 만들어보자.

```
\documentclass[a4paper,12pt]{article}
\usepackage{labels}
\LabelCols=3      % n. of columns of labels
\LabelRows=8      % n. of rows of labels
```

```

\LeftBorder=8mm    % borders of each label
\RightBorder=8mm
\TopBorder=5mm
\BottomBorder=5mm
\LabelGridtrue      % show the grid
\numberoflabels=10  % number of labels of each type to print
% the text of the label is specified by
% the \addresslabel[]{} macro:
\begin{document}
  \addresslabel[\large] % optional arguments
  {\textbf{Guido Gonzato}, Ph.D.\}
  \textsl{Linux system manager}}
% now on to the boxed labels
\boxedaddresslabel[\fboxsep=4mm\fboxrule=1mm]
{\textbf{Guido Gonzato}, Ph.D.\}
\textsl{Linux system manager}}
\end{document}

```

서로 다른 주소를 가진 레이블을 만들어야 한다면 외부 파일을 작성하든가 메인 파일에 주소를 써 넣든가 해야 한다.

```

\documentclass[a4paper,12pt]{article}
\usepackage{labels}
\LabelCols=3
\LabelRows=8
\LeftBorder=3mm
\RightBorder=3mm
\TopBorder=8mm
\BottomBorder=8mm
\LabelGridtrue
\begin{document}
% use either this environment:
\begin{labels}
  1$\{st\}$ name
  1$\{st\}$ address
  1$\{st\}$ city, state, zipcode

  2$\{nd\}$ name
  2$\{nd\}$ address
  2$\{nd\}$ city, state, zipcode

  3$\{rd\}$ name
  3$\{rd\}$ address
  3$\{rd\}$ city, state, zipcode
\end{labels}
% or an external file containing exactly the same text:
% \labelfile{addresses.dat}
\end{document}

```

textmerg와 labels 패키지를 함께 사용하는 방법은 스스로 시험해보라.

7.3 언어선택(Tools/Default Language)

TeX의 기본 언어는 영어이다. 그러나 다국어도 지원한다. 여기서 말하는 언어지원이라는 것은 “Chapter”나 “Index” 같은 용어를 “장”이나 “색인”으로 바꾸는 것, 하이픈 분리, 그리고 ‘ç’나 ‘é’와 같은 글자를 키보드에서 직접 써넣을 수 있는 기능을 말하는 것이다. (일반적으로 이 글자들은 \c c나 \’e와 같이 입력한다.)^{*xxviii}

TeX 배포판에 포함된 language.dat 파일에(t_ETeX에서는 \TeXMF/tex/generic/config/language.dat에 있다) 각 언어의 목록이 열거되어 있다. 이 파일을 편집하여 자신의 언어에 알맞은 하이픈 패턴을 선택할 수 있을 것이다.

영어 사용자가 아니라면 babel 패키지를 다음과 같은 식으로 사용할 수 있다.

```
\usepackage[italian,english]{babel}
```

또한, 첨자붙은 글자들이나 일반적인 비표준 ASCII 문자³를 (직접) 입력하려면, isolatin1 패키지를 사용할 수도 있다. 그러나, 이 패키지를 사용하는 것은 그다지 권장할 만한 일이 아니다. 왜냐하면 그렇게 하면 파일의 가독성과 이식성을 떨어뜨리기 때문이다. TeX의 보편적인 방식을 따르는 쪽이 더 낫다.

만약 한 글자를 입력하기 위해서 서너 개의 문자군을 쳐넣어야 하는 게 참을 수 없다면, 에디터 설정을 바꾸어서, 이 문제를 해결할 수 있다. 예컨대 나는 내가 주로 쓰는 jed 에디터로 하여금 내가 ‘é’를 입력할 때는 그것이 \’e로 바뀌어서 입력하도록 설정해두었다.

```
define latex_mode_hook ()
{
  set_abbrev_mode (1);
  if ( () = abbrev_table_p ("LaTeX") )
    use_abbrev_table ("LaTeX");
#ifdef WIN32
  % prevent clash with movement keys
  undefinekey ("àà", "LaTeX-Mode");
  definekey (" \\'a", "àà", "LaTeX-Mode");
#else
  local_setkey (" \\'a", "à");
#endif
  local_setkey (" \\'e", "é");
  local_setkey (" \\'e", "è");
  local_setkey (" \\'i{}", "ì");
  local_setkey (" \\'o", "ò");
  local_setkey (" \\'u", "ù");
}
```

^{*xxviii} 한국어에서는 하이픈이 없다. 그리고 특수문자의 입력은 주로 한글 코드에 관계된 것이므로, 비영어권 구미어 지원과는 다른 문제라고 할 수 있다.

³컴퓨터의 관습적 용어인 “표준 ASCII 문자”란 32 (공백)에서 126 (물결문자)까지를 포함하는 코드로 이루어진 글자를 가리킨다.

비슷한 일을 어떻게 할 수 있는가는 자신이 사용하는 에디터의 도움말을 참고하라.

7.4 자동 하이픈(Tools/Hyphenation)

TeX은 하이픈 처리를 아주 잘 해내는 편이다. 그렇지만 가끔 사용자가 직접 지정하는 쪽이 더 좋은 결과를 얻을 수도 있다. 사용자가 하이픈을 직접 지정하려면 \-를 단어의 하이픈이 일어나도록 하려는 위치에 써넣는다. 더 좋은 방법은 하이픈 규칙을 다음과 같이 선언하는 것이다.

```
\hyphenation{ge-o-phy-sics, ge-o-lo-gy, earth}
```

이렇게 설정하면 TeX은 “earth”라는 단어는 하이픈처리를 하지 않는다. 하이픈이 일어나지 않도록 하는 또하나의 방법은 단어 전체를 \mbox로 둘러싸는 것이다.

```
Do not hyphen \mbox{internationalisation}, please.
```

7.5 맞춤법 검사(Tools/Spell Check)

철자법 검사는 TeX이 관여하는 부분이 아니다.^{*xxix} ispell이나 aspell과 같은 외부 유틸리티를 이용해야 한다. UNIX에서는 ispell이라는 프로그램으로 다음과 같이 할 수 있다.

```
shell> ispell -t mydocument.tex
```

-t 스위치는 ispell로 하여금 TeX과 TeX 명령어에 대해서는 철자법 검사를 하지 않도록 지시하는 것이다. 영어 이외의 언어로 철자 검사를 하려면 -d 스위치를 사용해서 적절한 사전을 지시해 준다.

```
shell> ispell -d italiano -t mydocument.tex
```

제 8 절 도움말(Help) 메뉴

TeX에 관한 도움말을 얻을 수 있는 방법은 온라인이든 오프라인이든 여러 가지가 있다. 시작 위치로 가장 좋은 곳은 CTAN 사이트이다. <http://www.ctan.org/tex-archive/info/>.

- info latex (UNIX 시스템) 명령은 간단하지만 완전한 온라인 개요를 보여준다.
- <http://www.giss.nasa.gov/latex/> 이곳은 종합적인 온라인 참고문헌이다. 유용한 링크가 많다.
- <http://www.ctan.org/tex-archive/info/LatexHelpBook/> 이곳은 아주 좋은 TeX 도움말 시스템으로서, 윈도 시스템에서도 완벽하게 이용할 수 있다.

^{*xxix}현재 한글 TeX 문서에 관해서 맞춤법 검사를 하는 방법은 아직 신중한 방법이 알려져 있지 않다. 한국어의 경우 철자 및 구문 검사는 주로 한글이나 MS 워드와 같은 워드 프로세서에 내장된 기능으로만 구현되어 있고, 공개 프로그램으로 신뢰할 만한 성능을 가진 것은 역자로서는 알지 못하고 있다. 해결해야 할 문제 중의 하나라고 생각된다. 한국어 맞춤법 검사는 사용된 단어를 단순히 단어 목록에서 찾아서 대조하는 방법만으로는 충분하지 않기 때문이다.

- news:comp.text.tex 뉴스그룹도 빠뜨릴 수 없다. 도움을 얻을 수 있는 소중한 곳이다.

많은 GNU/Linux 배포판은 `teTeX`을 포함하고 있는데, 이것은 가장 완벽한 `TEX/LATEX` 시스템이다. 여기에 수많은 문서가 제공되고 있으며, 대부분 [TEXMF/doc](#) 디렉토리 아래에서 찾을 수 있다. 이 디렉토리를 훑어보기 바란다. 원하는 것을 대부분 얻을 수 있을 것이다.

제 9 절 마치는 말

© Guide Gonzato, 2001-2002. 이 문서는 GNU 일반 공개 라이선스로 배포한다. 이 지침서가 도움이 되기를 진심으로 바라며, 제안이나 논평은 환영한다. 언제라도 저자에게 연락하기 바란다.

부록 A 문서마당(Document Templates)

article 클래스에 해당하는 템플릿은 제 2.1 절에서 제시하였다. 다음에 보인 것은 더 많은 예제들이다.

```
\documentclass[twoside,11pt]{book}
\begin{document}
\frontmatter
\begin{titlepage}
\title{The Book of Mine}
\end{titlepage}
\author{John B. Smith}
\maketitle
\tableofcontents
\mainmatter
\part{The Beginning}
\chapter{Introduction}
\section{Let's Start}
The book starts here.
\part{The End}
\backmatter
Thank you for reading this book.
\end{document}
```

그림 3: Book template.

```
\documentclass[twoside,12pt]{report}
% tables and figures at the end:
\usepackage{endfloat}
\begin{document}
\title{Final Report}
\author{John B. Smith}
\date{London, \today}
\maketitle
\begin{abstract}
This is the final report.
\end{abstract}
\tableofcontents
\listoftables
\listoffigures
\part{Start}
\chapter{Begin}
\section{Introduction}
The report starts here.
\end{document}
```

그림 4: Report template.

```

\documentclass[12pt]{letter}
\begin{document}
\address{My address}
\signature{Guido}
\begin{letter}{John's address}
\opening{Dear John,}
Thank you for being my friend.
\closing{Hope to see you soon,}
\ps{P.S. Say hello to granny!}
\encl{My son's photographs!}
\end{letter}
\end{document}

```

그림 5: Letter template.

```

\documentclass[a4paper]{article}
\usepackage{type1cm}
\usepackage{times}
\usepackage{color}
\usepackage{rotating}
\pagestyle{empty}
\begin{document}
\begin{sidewaysfigure}
\fontsize{2.5cm}{2.5cm}\selectfont
\centerline{\textcolor{blue}{\textbf{Please:}}}}
\vskip 1cm
\fontsize{4cm}{3cm}\selectfont
\centerline{\textcolor{red}{DO NOT}}
\centerline{\textcolor{red}{SMOKE}}
\centerline{\textcolor{red}{HERE!}}
\vskip 1cm
\fontsize{2cm}{2cm}\selectfont
\centerline{\textcolor{magenta}{If you do,}}
\centerline{\textcolor{magenta}{you'll be \emph{deboned!}}}}
\end{sidewaysfigure}
\end{document}

```

그림 6: How to write a notice.

```

\documentclass{article}
\usepackage[absolute,showboxes]{textpos}
\usepackage{color}
\usepackage{framed}
\usepackage{graphicx}
\setlength{\TPHorizModule}{10mm} % standard unit of length
\setlength{\TPVertModule}{\TPHorizModule}
\setlength{\TPboxrulesize}{1pt} % box line width
% start everything near the top-left corner
\textblockorigin{0mm}{0mm}

\begin{document}
\setlength{\parindent}{0pt}
\definecolor{shadecolor}{rgb}{0.9,1,1}
\begin{textblock}[5](0,0)
% this block is 5 modules wide; height is
% automatically determined
\begin{center}
\begin{minipage}[c]{0.8 \linewidth}
\begin{shaded}
This block is placed with its top left corner at the 'origin'
on the page, which has been set to (0mm,0mm). The internal
margin and the shading are provided by the \texttt{minipage}
and \texttt{shaded} environments.
\end{shaded}
\end{minipage}
\end{center}
\end{textblock}
\begin{textblock}[6](10,1)
\includegraphics[width=6cm,angle=-90]{gnuplot.ps}
This picture is at (10,1). Note that rotating it
by -90 makes it overflow the margin.
\end{textblock}
\begin{textblock}[5][0.5,0.5](2.5,8)
This block is at position (2.5,8), but because the optional
argument [0.5,0.5] has been given, it is the centre of the block
which is located at that point, rather than the top-left corner.
\end{textblock}
\begin{textblock}[3,4](6,4)
The dimensions of this block are 3$\times$4 cm.
Its origin is position (6,4) on the page. Note that the text
overflows the margin in some cases; you'll want to
use the \texttt{minipage} environment to prevent that.
\end{textblock}
\end{document}

```

그림 7: How to write a poster.