

# 목차

I	개관	1
1	TEX이란 무엇인가?	3
1.1	TEX의 탄생	3
1.2	TEX의 동작원리	5
1.3	TEX의 장점과 단점	6
1.4	누가 TEX을 사용하는가?	11
2	원레이텍은?	13
2.1	원레이텍 1.0 사양	15
2.2	설치	17
2.2.1	시스템 환경	17
2.2.2	설치	17
3	원레이텍 동작 입문	19
4	주프로그램 설명	25
4.1	화일메뉴	26
4.2	편집메뉴	29

4.3	검색메뉴	30
4.4	입력메뉴	31
4.5	조판메뉴	34
4.6	창메뉴	35
4.7	도움말 메뉴	35
4.8	HTML 화일에서 $\LaTeX$ 화일로 읽어들임	37
4.9	$\LaTeX$ 화일로 부터 HTML 화일로의 변환	39
4.9.1	왜 HTML로의 변환이 필요한가?	39
4.9.2	HTML 변환기의 기능	40
4.9.3	도스 코맨드 라인에서 변환기의 수행	42
4.9.4	테이블및 그래픽스의 사용	44
4.9.5	앞으로의 버전에 대한 계획	47
5	DVIWIN 설명	49
5.1	DVIWIN 이란?	49
5.2	화면 움직이기	50
5.3	그림 넣기	50
5.4	인쇄하기	53
5.5	$\TeX$ 및 기타 프로그램과의 관계	55
5.6	메뉴 설명	56
5.7	참고 사항	62
6	원레이텍 포맷 설명	63
6.1	$\TeX$ 에서의 한글 윈도우의 한글 폰트이용	63
6.2	원레이텍의 $\LaTeX$ (lplain)	66
6.2.1	한글 counter	66

II	LaTeX 설명	69
7	QUICK START: 한글 LaTeX 화일의 예	71
7.1	짧은 글	73
7.1.1	한 두가지 경고	73
8	글자입력	79
8.1	글자모양	79
8.2	글자크기	80
8.3	섞어 쓰기	80
8.4	특수기호의 출력	81
9	수식 쓰는 방법	85
9.1	간단한 수식 표현	85
9.2	복잡한 수식 표현	88
9.3	배열(array)	90
9.4	기호및 기타	92
9.5	수식의 사용의 실례	93
9.6	수식출력의 예	93
9.7	월리스의 곱	97
9.7.1	기초적인 수식	98
9.7.2	수학기호	98
10	Table과 Tabbing	101
10.1	tabbing 환경	101
10.2	tabular 환경 { 표 그리기	104
10.3	Table과 Figure 환경	106

11	여러가지 문단 정렬 방법	109
11.1	center, .ushleft, .ushright	110
11.2	itemize, enumerate, description	111
11.2.1	Custom Description Lists	112
11.3	quote, quotation	114
11.4	verse { 시를 쓸때	115
11.5	verbatim과 verb	117
12	큰 글 짜기	119
12.1	section 명령어	119
12.2	Title and Titlepage	120
12.3	목차 { Table of Contents	121
12.3.1	머릿말	122
12.4	Preamble	122
12.4.1	Document Style Options	123
12.4.2	단면 그리고 양면 출력	123
12.4.3	header	124
12.4.4	Page Layout Option	125
12.5	Appendix { 부록	126
12.6	큰 문서작성	126
13	Bibliography의 작성	131
13.1	Bibliography Database 파일 : BIB 파일	131
13.2	Entry Types	132
13.3	Field	134
13.4	텍스트에 reference 달기	135
13.5	BibTEX을 이용해 bibliography 만들기	136

13.6	BibTEX없이 bibliography 만들기	137
14	Glossary와 Index 만들기	139
14.1	Glossary	139
14.2	Index	140
14.2.1	index 할 단어의 mark	140
14.2.2	MakelIndex를 사용한 index의 생성	143
15	Two-Column Document	145
15.1	Twocolumn 옵션	145
15.2	Twocolumn Proceeding 옵션	145
15.3	텍스트 중에서 two-column으로 전환	146
15.4	Minipage environment	146
16	여러가지 유용한 스타일 화일	151
16.1	multicol.sty	152
16.2	fancybox.sty	152
16.3	verbatim.sty	153
17	특수 기능	155
17.1	Newcommand, Renewcommand를 사용한 Macro의 생성	155
17.2	Newenvironment, Newtheorem	156
17.3	Counter	158
18	Error	161
18.1	자주 발생하는에러	161
18.2	에러를 어떻게 고칠것인가.	162
18.3	경고	162

III	보기화일 Samples	163
19	Samples 화일	165
19.1	바자회 알림	165
19.2	동창회 정관	166
19.3	수학기호	168
19.4	논문	168
19.5	책소개	170
A	수학기호	175
B	참고 문헌 설명	181

제 1 부

개관





# 1장

## TEX이란 무엇인가?

### 1.1 TEX의 탄생

TEX은 수학자겸 전산과학자로 유명한 Stanford 대학의 Donald E. Knuth 교수에 의해서 만들어졌다. Knuth 교수의 역작인 The Art of Computer Programming의 부산물로 TEX이 나오게 되었다. TEX의 탄생에 관한 이야기를 Knuth 교수의 말로 들어보자:

왜 나는 1977년에 TEX을 만들기 시작했는가? 실제로 전체적인 일은 The Art of Computer Programming 책과 연관되어 이미 오래전에 시작되었다. 그 당시 나는 Volume 2의 2판을 준비했었는데, 1판을 출판한 이후로 인쇄기술이 급격하게 변하여 출판사로부터 형편없는 교정판을 받게되었다. 교정판은 남으로 식자(typesetting)하는 대신에 사진식자(phototypesetting)로 되어있고 사람손 대신에 컴퓨터의 도움을 받아 제작되었다. 그결과는 수식에서 간격이 엉망이었

고, 폰트는 원본에 비해서 형편없었다. 나는 이것에 크게 실망하여 어떻게 해야될지를 몰랐었다. Addison-Wesley사는 다시 납으로 식자해줄 것을 제안했으나 나는 그러한 구식 방법은 머지않아 곧 사라질 것을 알고 있었고 내가 받은 교정판처럼 보이는 책을 쓰고 싶지는 않았다.

그리고 ::: 우리는 남가주(southern california)에서 새로운 기계로 만든 Pat Winston의 Arti- cial Intelligence의 교정판을 받아 보았는데, 모든 것이 고해상도 래스터에 근거를 하고 ::: 디지털 글자는 나의 교정본의 글자보다 훨씬 보기 좋았고 ::: Winston의 교정본을 보고나서 일주일 후에 나는 모든 것을 취소하고 디지털 인쇄에 전념하기로 결심했다.

1977년에 이렇게 시작한 이후로 TEX 연구 프로젝트는 두 가지 목표를 갖고 추진되었다. 첫번째 목표는 질(quality)이었다. 우리는 단순히 좋은 것 이상의 최상의 문서를 만들기를 원했다 ::: 1977년도 당시만해도 매우 매력적인 문서를 만들 수 있는 시스템이 몇 개가 있었다. 나의 목표는 거기서 더 나아가 인쇄된 문서가 얻을 수 있는 최고의 질을 얻는 것이었다.

약 2년간의 작업끝에, 텍스트 포매팅에 관한 한 이정도 수준의 질을 얻는 것은 그리 어렵지 않다는 것이 밝혀졌다. 예를 들어, 우리는 TEX으로 Time 잡지를 원래의 Time 잡지보다 보기 좋게 만들 수 있다는 것을 증명했다. 그러나 글자를 만들어 내는 것이 우리가 예상한 것 보다는 훨씬 어렵다는 것을 알게 되었다. 내가 좋아하기 시작한 글자형을 만드는데 7년이 걸리게 되었다.

두번째 목표는 기록용(**archival**)이었다: 가능한한 인쇄기술의 변화에 크게 의존하지 않는 독립적인 시스템을 만드는데 목표가 있었다. 다음 세대의 인쇄 장비가 도래해도 새로이 모든 문제들을 해결하는 대신에 이전에 기록해 놓은 질을 유지하고 싶었다. 나는 100년이 지나도 계속 사용 가능한 무엇인가를 디자인하고자 했다. 다른 말로 하면, 만일 책의 사양이 현재의 상태로 저장된다해도 2086년에 우리의 후손들이 만들어도 지금 상태의 것과 똑같은 책을 만들 수 있게 하는 것이 나의 목표이었다 :::

## 1.2 TEX의 동작원리

일반적으로 텍스트 처리기(**processor**)는 크게 두가지로 나눌 수 있다:

- 0 **WYSIWYG<sup>1</sup> 시스템**: 인쇄될 문서의 모양이 화면에 그대로 나타나고 입력을 하면 그 변화가 즉시 화면에 반영되어 나타난다.
- 0 **마크업(markup) 시스템**: 텍스트와 포매팅 명령어를 섞어 입력하는데 그 효과를 바로 보지는 못한다. 프로그램을 수행하여 입력된 내용을 처리하면 결과 이미지를 화면이나 프린터로 볼 수 있다. 컴퓨터 용어로 표현하면 마크업 시스템은 입력한 *소스 화일*을 *킵파일*해야한다.

**WYSIWYG** 시스템은 바로 피드백이 되므로 분명히 장점이 있으나, 아주 정확하지는 않다: 예를 들어, **WYSIWYG** 시스템으로 문서를 만들어 300 DPI(Dots Per Inch) 프린터로 인쇄하여 보면 적당히 좋아 보이

---

<sup>1</sup>What You See Is What You Get. 화면과 프린터가 같은 해상도, 서체 그리고 페이지 언어를 사용하면 100% **WYSIWYG**이다.

나 고해상도로 식자를 하면 더이상 좋게만 보이지는 않는다. 인간의 눈은 매우 민감하다: 마치 오케스트라 연주시 누군가가 작은 실수를 하면 정확히 무엇이 잘못되었는지 지적을 할 수는 없으나 귀에 거슬리는 것과 마찬가지로 잘못 인쇄된 문서는 눈에 거슬리게 된다. 실제로 아름답고, 알아보기 쉽고, 그리고 편히 읽을 수 있는 문서를 만들기 위해서는  $\frac{1}{1000}$  인치내에서 정확히 문서의 모든 요소들이 위치해야 한다. 이러한 유형의 작업에는 즉시 피드백되는 것이 그리 장점이 되지 않는다: 간격, 정렬 등의 세밀한 조정은 저해상도 화면을 통해서 하기에는 너무 힘이 들고 손으로 일일이 작업하는 것이 불가능한 경우도 있다.

이러한 이유로 인해 전문적으로 식자를 하는 곳에서는 마크업 시스템이 환영을 받는다. 마크업 시스템을 사용하면 정밀한 작업을 명령어로 처리하고 자동화 시킬 수 있다. 그러나 이러한 접근 방식은 결과를 바로 볼 수가 없기 때문에 초보자에게는 매력적으로 보이지 않게 된다. 따라서 WYSIWYG 입력기가 포함된 마크업 시스템을 사용하면 양쪽의 장점을 모두 얻을 수 있을 것이다.

TEX은 마크업 시스템에 속한다. 즉, 사용자는 문서의 내용과 필요한 명령어가 들어간 소스 파일을 TEX 엔진을 이용하여 컴파일하여 인쇄를 하거나 화면을 통해서 결과를 보게 된다. 컴파일된 결과로 나온 파일을 DVI(DeVice Independent) 파일이라고 하는데 화면이나 프린터의 해상도에 관계없이 드라이버만 있으면 결과를 볼 수 있다.

### 1.3 TEX의 장점과 단점

무엇보다도 TEX의 장점은 높은 질의 문서<sup>2</sup>를 만들어 낸다. TEX이 다른 문서 시스템에 비해 텍스트 처리에서 특히 뛰어난 몇가지 예를 보면 다

---

<sup>2</sup>특히 수식의 경우

음과 같다:

- 0 합자(ligature)<sup>3</sup>를 자동으로 처리한다.
- 0 커닝(kerning)<sup>4</sup>을 자동으로 처리한다.
- 0 정판(justification)을 하는 복잡한 체계를 갖고 있어 필요하면 하이프네이션<sup>5</sup>을 한다. 하이프네이션 규칙은 사용자가 원한다면 추가나 변경이 용이하여 여러 언어에서 사용 가능하다.
- 0 수학 공식내의 여러 요소들 사이의 간격은 최고의 수식 식자공들이 사용하는 전통적인 규칙에 따라 **TEX**에 의해서 결정되기 때문에 수식이 아름답게 처리된다.

잘 씌어진 **TEX**문서는 매크로에 의해서 포맷되어 문서의 각 요소들이 어떻게 식자되는지 지정된다. 매크로는 기본적인 포맷팅 명령어들로 만든 짧은 프로그램으로서 일일이 세세한 명령어를 사용하는 번거로움을 덜어준다. 예를 들어, `\footnote` 매크로는 텍스트를 페이지의 바닥으로 이동시키거나 줄 위에 작은 번호들을 매기는 걱정을 하지 않아도 자동으로 주석 처리가 되게 한다. 매크로들은 `\indent` 혹은 `\par`와 같은 기본 명령어(primitive)로 씌어져 **TEX**의 기본적인 어휘가 된다.

매크로 덕분에 **TEX**은 대단히 다양한 용도의 기능을 갖게 되었다. 예를 들어, 문서의 외관을 고치고자 하면 텍스트를 건드리지도 않고 단순히 특정 매크로의 정의만 고쳐도 충분하다. 온라인 문서의 코딩에 관한 표준을 정의하는 작업이 현재도 진행중인데, **TEX**은 고급(high-level) 문

---

<sup>3</sup>영어의 `..`, `...`, `/`, `0` 등과 같이 여러 글자가 합쳐져 하나의 글자로 처리된다. 전통적으로 합자는 시각적으로 보기 좋은 결과를 얻기 위해 필요하다.

<sup>4</sup>글자의 모양이 서로 가능하면 가깝게 위치하게 하여 글자사이의 간격이 균등하게 보이게 한다.

<sup>5</sup>단어의 중간에 하이픈(-)을 삽입하여 줄사이에서 끊어지게 한다.

서 표현 언어(document description language)로 사용될 수 있기 때문에 쉽게 이 표준에 들어갈 것이다.

TEX은 이동이 가능(portable)하다. 텍스트와 매크로를 포함한 TEX으로 씌어진 문서는 인쇄 가능한 ASCII 세트로 코드될 수 있다. 따라서 모뎀이나 LAN 같은 통신 수단을 이용하여 아무리 복잡한 수식이나 레이아웃(layout)이 포함된 문서라도 세계 어느 곳에서도 서로 주고 받을 수 있다. 예를 들면, 미국 수학회(American Mathematical Society)에서 보유하고 있는 데이터베이스는 TEX에 근거를 하여 세상에 출판된 모든 수학 논문의 요약(abstract)들을 포함하고 있다. 따라서 원하는 논문의 요약을 데이터베이스에서 TEX으로 포맷팅된 형태로 얻을 수 있다.

TEX은 특정 컴퓨터의 특성에 의존하지도 않고 기계와는 독립적인 방법으로 계산을 하기 때문에 여러 컴퓨터들 사이에서 이동 가능하다. 즉, Macintosh, PC, UNIX 워크스테이션, IBM 대형 컴퓨터, 심지어는 Cray등에서 각각 만든 TEX문서는 모두 모양이 같게 된다. 따라서 여러 컴퓨터 시스템 사이에서 정보교환의 수단으로 TEX은 매우 적합한 도구가 된다.

TEX은 문서의 이미지를 직접 만들어 내는 대신에 단지 페이지의 내용을 묘사한다. 문서를 인쇄하기 위해서는 TEX의 출력<sup>6</sup>을 받아 이를 드라이버 프로그램으로 원하는 프린터나 화면에 출력해야 한다. 이러한 이유로 인해 TEX은 현재의 프린터 기술과도 독립적이 된다. 만일 프린터 기술이 변한다면 문서의 내용을 바꿀 필요없이 단지 필요한 드라이버 프로그램만 만들면 된다.

TEX은 단순한 텍스트 프로세서 이상이다. 프로그램 언어이다! 사용자가 원하는 다양한 기능을 TEX의 변수를 변경하거나 새로운 매크로를 정의함으로써 추가할 수 있다. 경험이 많아지면, 새로운 스타일을 정

---

<sup>6</sup>DVI 화일

의하고 특정 목적을 위한 복잡한 매크로를 만들 수도 있고 이미 이전에 남들이 만들어 놓은 스타일이나 매크로를 활용할 수도 있다. 물론 이러한 일들이 쉽지만은 않지만 TUG(TeX User Group)을 통해서 많은 정보를 얻을 수 있으며 네트워크를 통해 public domain에 나와있는 정보나 데이터를 활용하면 많은 도움이 될 것이다.

TeX은 또한 확장 가능하대 언급했듯이 Knuth는 TeX을 만들 당시에 미래를 바라보았었다. 이러한 이유로 그는 TeX을 양파와 같은 층구조로 만들었다. 중앙에는 TeX의 300개의 기본 명령어(primitive)로 되어 있는데 이름 그대로 원시적이어서 실제로는 기본 명령어를 바로 사용하는 경우는 적다. 그다음 층은 이보다 고급 명령어들 혹은 매크로들로서 TeX의 포맷 화일에 정의되어 있다. 가장 보통 사용되는 포맷 화일은 plain.tex 으로 약 600개<sup>7</sup>의 명령어들이 정의되어 있다. 기본 명령어와 확장 명령어의 조합을 Plain TeX이라고 일반적으로 불리운다. 이위에 확장을 한 층을 패키지라고 하는데 대표적으로는 LaTeX, AMSTeX 등이 있다. 이중에 LaTeX이 가장 알려져 있으며 가장 완벽하다. 따라서 대부분의 TeX사용자는 실제로는 LaTeX을 쓰고 있다.

TeX은 매우 잘 디버그(debug)되어 있다. 물론, 다른 프로그램들과 마찬가지로 결코 버그(bug)가 없지는 않다. 그러나 새로운 버그가 보고될 때 마다 Knuth가 상을 준 덕(?)에 전세계적으로 버그 사냥꾼들이 몰려들어 극히 유별나거나 즉시 발생할 사유가 없는 버그들만 남아있다. 만일 지금 이라도 버그를 발견하여 보고하면 상과 함께 TUGboat에 정기적으로 출판되는 공식적인 TeX버그 목록에 실리게 된다.

TeX은 public domain에 나와 있다. Knuth는 TeX을 세상에 공짜로 제공했다. TeX의 소스 프로그램은 쉽게 복사할 수 있다. 만일

---

<sup>7</sup>이 숫자에 놀라지는 마라. 왜냐하면 이중에 알아야 할 것은 100개도 안되며 그것도 암기하기 쉬운 이름으로 정의되어 있다

TEX을 산다면, 이는 특정 기계 및 환경에 설치된 드라이버, 보기 프로그램(previewer), 텍스트 에디터 등에 대해서 지불하게 되는 것이다.

완벽한 것은 없다. TEX의 단점의 대부분은 확장 혹은 보조 프로그램을 통해서 극복될 수 있다. 단점중에는 특정 설치환경에 기인하지 TEX의 본질적인 단점이 아닌 것도 있고, TEX의 보급 초기에 제기되어 이미 여러 방법으로 해결된 것<sup>8</sup>도 있다.

TEX 프로그래밍은 난해하여 습득하는데 시간이 걸린다. 물론 주어진 명령어나 매크로를 사용하는 일반 사용자에게는 이것이 문제가 되지는 않는다. 그러나 새로운 매크로나 스타일을 디자인하는 것은 초보자에게 적합한 과제가 아니다.

TEX은 제한된 메모리를 갖고 있다. 초기에 설치된 TEX들은 메모리의 한계 때문에 사용할 수 있는 매크로나 서체의 숫자가 제한되었으나 컴퓨터 OS(Operating System)의 발전으로 이를 극복한 환경<sup>9</sup>들이 계속 나오고 있다.

TEX은 대화형식(interactive)이 아니다. 이 점이 TEX의 최대 단점으로 일반적으로 지적되고 있다. TEX은 본질적으로 마크업 시스템이기 때문에 이 문제를 완전히 해결할 수는 없지만 사용하기 쉬운 텍스트 에디터, 보기 프로그램, 프린터 드라이버 등이 잘 합해져 사용자가 마크업 시스템에서 얻기 힘든 피드백을 쉽고 빠르게 받을 수 있다면 극복될 수 있다.

---

<sup>8</sup>서체 추가, 그래픽 처리, 다양한 레이아웃

<sup>9</sup>X-Window, PC Windows, Mac OS



## 1.4 누가 T<sub>E</sub>X 을 사용하는가?

Knuth가 T<sub>E</sub>X을 만든 첫번째 이유는 자기 자신이 필요했기 때문이다. 즉, Knuth와 같이 학문을 하거나 연구를 하는 사람들이 출판을 하는 경우 T<sub>E</sub>X을 사용하는 것이 최선의 선택이 된다. Knuth는 자신이 필요한 것이 무엇인지를 알고 있었고 그것을 T<sub>E</sub>X이 충실하게 실현하도록 만들었다. 따라서 T<sub>E</sub>X은 전세계적으로 학계나 연구기관에서 출판 혹은 정보교환용으로 표준으로 사용되고 있으며, 문서 데이터베이스의 구축이나 온라인 문서의 기록(archive)에 활용되고 있다.

분야에 관계없이, 글을 쓰는 사람은 T<sub>E</sub>X의 사용을 한번 고려해 보아야 한다. 글을 쓰는 사람은 글의 모양<sup>10</sup>보다는 내용에 신경을 써야한다. 컴퓨터의 발달과 이에 따른 보급의 확장으로 대부분의 글을 쓰는 사람들은 워드프로세서를 사용하여 글을 입력한다. 이렇게 입력된 글은 직접 프린터로 인쇄하거나 출판사에 의뢰하여 식자가 되어 출판을 된다. 이때 글의 모양은 글을 쓰는 사람에 의해서 결정되는 경우가 많고 심지어는 글의 모양을 스스로 책임을 져야하는 경우도 있다. 글을 쓰는 사람들은 실제로 전문 식자공이 아니기 때문에 읽기 쉽고 아름다운 문서를 만드는데 어려움이 있으며 이때문에 많은 노력을 내용과는 상관없는데 기울이게 된다. 만일 원하는 모양이 복잡한 레이아웃도 아니고 반복하여 계속 사용하는 경우에 T<sub>E</sub>X을 사용하면 글의 내용만 신경을 쓰면 되니까 훨씬 더 좋은 글을 쓰게 된다.

책의 출판은 다른 문서의 출판과는 다른 점들이 있다:

- 0 내용이 길기 때문에 하나의 화일로 입력이 불가능한 경우가 많으며 여러 사람이 입력을 해야되는 경우도 있어 책의 전반에 걸쳐 모양의 일관성(consistency)을 유지하기 어렵다.

---

<sup>10</sup> 레이아웃, 서체의 지정 etc.

0 새로운 내용의 추가나 변경으로 전체적인 변화를 요구하는 경우<sup>11</sup>가 많다.

0 페이지 내용에 반복되는 디자인 요소들이 많다.

위에 열거된 문제점들은  $\text{TeX}$ 의 사용을 통해 해결될 수 있기 때문에 책과 같이 긴 문서의 출판에는  $\text{TeX}$ 의 사용이 적절하다.

$\text{TeX}$ 의 장점중의 하나는 수식처리가 쉽고 결과가 아름답게 나온다는 점이다. 이 때문에 수식을 많이 사용하는 이공계와 과학계통의 출판물은  $\text{TeX}$ 을 많이 사용하고 있다.

외국어<sup>12</sup>가 포함된 문서의 작성시도  $\text{TeX}$ 의 사용이 불가피하다.

---

<sup>11</sup> 색인, 목차등의 페이지 번호, 수식 번호, 참고 문헌, 주석 *etc.*

<sup>12</sup> 독일어, 불어, 희랍어, 라틴어, 스페인어 *etc.*

## 2장

# 윈레이텍은?

윈레이텍은 한글 윈도우 운영 체제에서  $\text{TeX}$ 을 사용하기 위하여 통합된 여러가지 프로그램을 지칭한다.

동일한  $\text{TeX}$ 엔진위에 한글 조합형의 코드 변환을 미리 처리하여  $\text{TeX}$  매크로로서 한글을 제공함으로 영문  $\text{TeX}$ 에서 만들어진 많은 매크로 시스템 (예: Plain  $\text{TeX}$ ,  $\text{\LaTeX}$  or  $\text{Am\LaTeX}$ )을 사용할 수 있다.

Win $\text{TeX}$ 는 한글 윈도우의 한글 폰트와  $\text{TeX}$ 의  $\text{CMR}$ (Computer Modern Roman) 서체를 True Type로서 제공함으로써 적은 디스크 용량으로 윈도우에 설치된 어떤 출력기에도 인쇄할 수 있다.

Win $\text{TeX}$ 는 다음과 같이 구성되어 있다.

1. Win $\text{TeX}$ 입력 에디터
2.  $\text{TeX}$ 엔진 : g $\text{TeX}$  2.1
3. 매크로: plain  $\text{TeX}$ ,  $\text{\LaTeX}$ ,  $\text{Am\LaTeX}$
4. DVI 보기(보기및 인쇄용)

## 5. True Type 영문서체

## 6. 예제 화일

## 7. hlx2tex 변환기

입력화일은 인쇄하고자 하는 문서의 내용과 **TEX** 명령을 포함하고 있는 화일이다. 이는 **ASCII** 형식으로 보통 확장명을 **.tex**으로 하는 경우가 많다. 한글을 사용할 경우에는 반드시 확장자를 **.hlx**로 하여야 한다. 이는 한글**TEX**이 한글이 포함된 **.htx**화일을 보통의 **TEX**이 처리할 수 있는 **.tex**화일로 변환하기 때문이다.

한글 **TEX**은 저장된 화일을 **TEX** 형식으로 바꾸어주는 과정을 거친 후에 이를 다시 화면이나 프린터에 출력하기 위한 화일로 바꾸어준다. 이 과정에서 몇개의 임시 화일을 만든다. 각 화일의 의미는 다음과 같다.

- 0 .hlx : 한글이 포함된 **LaTEX** 화일
- 0 .htx : 한글이 포함된 **TEX** 화일
- 0 .tex : 영문 **TEX** 화일 혹은 hlx 화일에서 **TEX** 형식으로 바뀐(translated) 화일
- 0 .dvi : **screen**이나 **printer**로 출력하기 위한 화일
- 0 .aux : 목차나 색인을 만드는데 필요한 내용이 들어있는 화일
- 0 .bak : backup 화일

**hlx2tex**는 **.hlx**화일을 **.tex** 화일로 변환시킨다. 주기능은 2바이트 한글 코드를 **TEX**에서 깨지지않는 문자토큰으로 간단한 매크로 함수로 만들어 주는 역할이다.

TEX이나 L<sup>A</sup>T<sub>E</sub>X의 기능으로 \input{}나 \include{}로 다른 파일을 포함하는 문자열이 나타났을때, 해당파일의 자동변환을 수행한다. 단 %로서 설명문장줄 안의 \input{}나 \include{}가 지시하는 파일은 변환하지 않는다.

h<sub>l</sub>x2<sub>t</sub>ex는 .h<sub>l</sub>x 파일을 .t<sub>e</sub>x 파일로 변환시킨다. .h<sub>l</sub>x 파일 안에 \input{}된 파일은 recursive하게 자동변환을 수행한다. 단 %나 \begin{verbatim} 환경안에 있는 \input{} 매크로는 변환하지 않는다.

## 2.1 윈레이텍 1.0 사양

### TEX 프로그램

0Original WEB -le authored by Donald Knuth, 0Karl Berry's WEB2C ver. 6.1, 0Ryu's Window version gTEX2.1

### 한글 L<sup>A</sup>T<sub>E</sub>X 포맷

: 0 based on L<sup>A</sup>T<sub>E</sub>X version 2.09 authored by Leslie Lamport, 0 supporting verbatim 0type change by logical font name of latex, with Hangulized Style -les (hbook, harticle, hreport..) and 0 many samples -les: tt 시험문제 (화학, 영어, 국어, 수학), Fliers (생일초대, 마자회, 책소개), MusicTeX(dvi only), 논문형식, 수식표기, 동창회 정관, 주소록 이력서

### DVI 보기 프로그램

0 Use of both PK fonts and True Type CMR FONT. and using 한글 Window's TrueType Font) 0 Fast display using internal fonts cache. 0 Global Zoom in/ out 0 Magnifying Glass 0 Resolution upto 600dpi 0 \special : Picture Display (wmf, pcx....) 0 Memorize of latest four -les. 0 Various paper selection menu

## Editor환경

0 Unlimited File size for editing  
0 Unlimited UNDO, REDO and REPEAT function  
0 Global FIND/REPLACE MDI (Multiple Document Interface)  
0 명령어 Tool Bar (T<sub>E</sub>X command and dvi preview)  
0 Synchronizing Text Scrolling options  
0 Printer font selection for display.  
0 Memorize nine history -le of latest edition.  
0 Triple Click LINE selection  
0 Quad Click paragraph selection  
0 Edit area right click menu  
0 LOG -le view command after T<sub>E</sub>X compilation.  
0 Multiple -le selection in opening -les dialog.  
0 Edition menu and Dialog Box for inserting latex command.  
0 Floating toolbar for latex command and character inputs.  
0 Import ASCII -le  
0 Launching programs : l a t e x, h l x, h t x  
0 Pulldown menu : w b r, c l i p n e t a, c h a r m a p  
0 File extension dependent T<sub>E</sub>X compilation  
0 Scrolling Windows New File Template selection.  
0 Automatic memory compacting before compilation  
0 Maximize DVIWIN when the module is resident and invoked.

## HTML , l a T E X 변환환경

0 Internet Home page printing by l a T E X i m p o r t  
0 Internet WWW publishing by HTML export from l a T E X including table of contents page and Browse control anchor.

## 2.2 설치

### 2.2.1 시스템 환경

원레이텍의 설치/ 운영을 위해서는 다음과 같은 시스템 환경이 갖추어 있어야 한다.

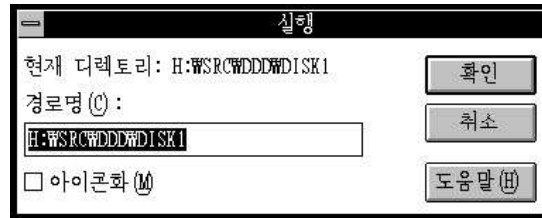
- 0 IBM 386 이상의 PC(486 추천)
- 0 한글 윈도우 3.1 이상
- 0 8MB 이상의 RAM(16MB 추천)
- 0 VGA 이상의 모니터

### 2.2.2 설치

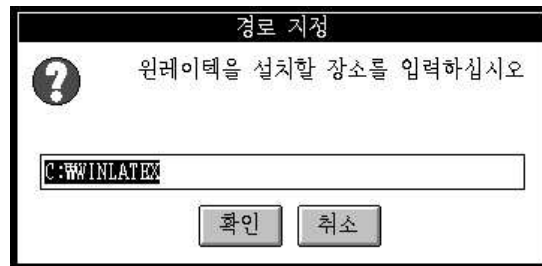
원레이텍의 설치는 다른 윈도우 프로그램의 설치와 유사하다.

1. 한글 윈도우를 올린다.
2. 원레이텍 디스크 1을 드라이브 A: 혹은 B:에 넣는다.
3. 프로그램 관리자 윈도우 메뉴에서 화일을 선택한다.

4. 화일 메뉴에서 실행을 선택하여 A:SETUP.EXE 혹은 B:SETUP.EXE를 입력하고 Enter 키를 누르면 설치 프로그램이 실행된다.



5. 윈레이텍이 설치될 경로(path)를 물어보는 다이얼로그 박스가 나타나면 원하는 경로명을 입력한다.



6. 화면에 나타나는 설치안내문에 따라 해당하는 디스크를 차례대로 A: 혹은 B:에 넣는다.
7. 설치가 끝나면 윈레이텍 프로그램 그룹과 아이콘이 등록된다.
8. 한글 윈도우를 재시동한다.



## 3장

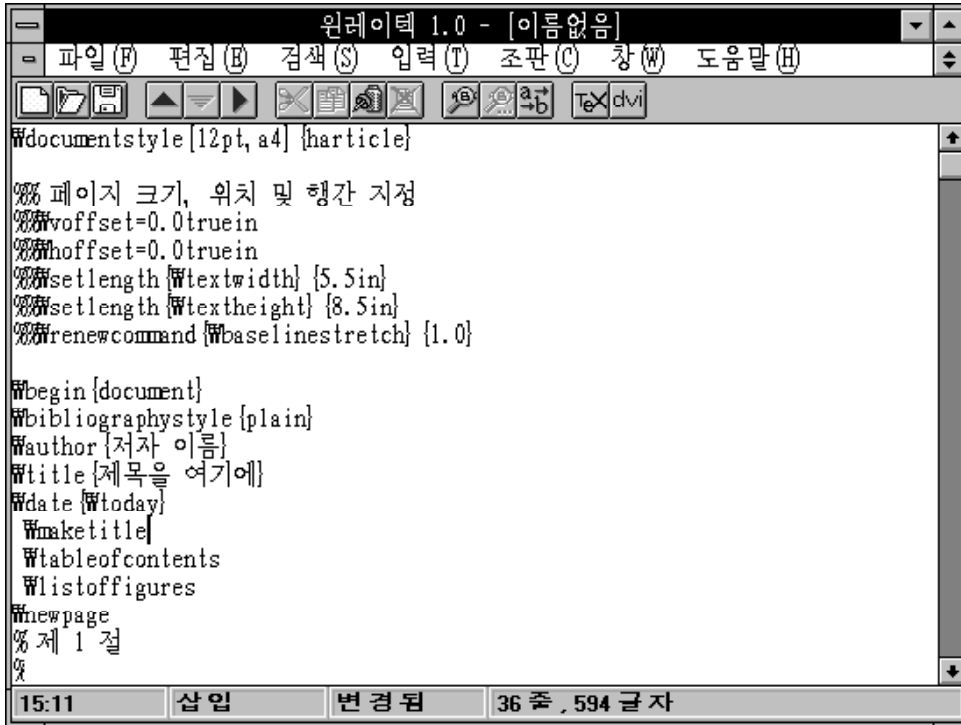
# 원레이텍 동작 입문

프로그램을 가장 빨리 배우는 방법은 직접 사용해 보는 것이다. 일반적으로  $\text{T}\text{E}\text{X}$  사용자의 대부분은  $\text{T}\text{E}\text{X}$  그자체 보다는  $\text{L}\text{A}\text{T}\text{E}\text{X}$ 을 사용하기 때문에 이장에서는  $\text{L}\text{A}\text{T}\text{E}\text{X}$  화일의 작성, 컴파일, 그리고 `preview` 및 인쇄에 대해서 간단하게 알아보기로 한다.

원레이텍이 제대로 설치 되었으면 프로그램관리자 윈도우내에 원레이텍 프로그램 그룹이 생성되어 원레이텍 주프로그램, `DVI` 보기, `Clipmeta`, `MathType` 등의 아이콘(icon)들이 등록된다. 각각의 아이콘은 독립된 프로그램들로서 필요에 따라 독립적으로 실행할 수도 있고, 원레이텍 주프로그램에서 나머지 프로그램들을 구동시킬 수도 있다. 보통은 원레이텍 주프로그램에서 텍스트 화일을 만들고, 이를 컴파일하고, `preview` 및 인쇄를 하는 것이 생산적이다. 마우스로 원레이텍 아이콘을 두번 클릭하면 주프로그램이 실행되고 초기화면이 다음과 같이 나타난다.



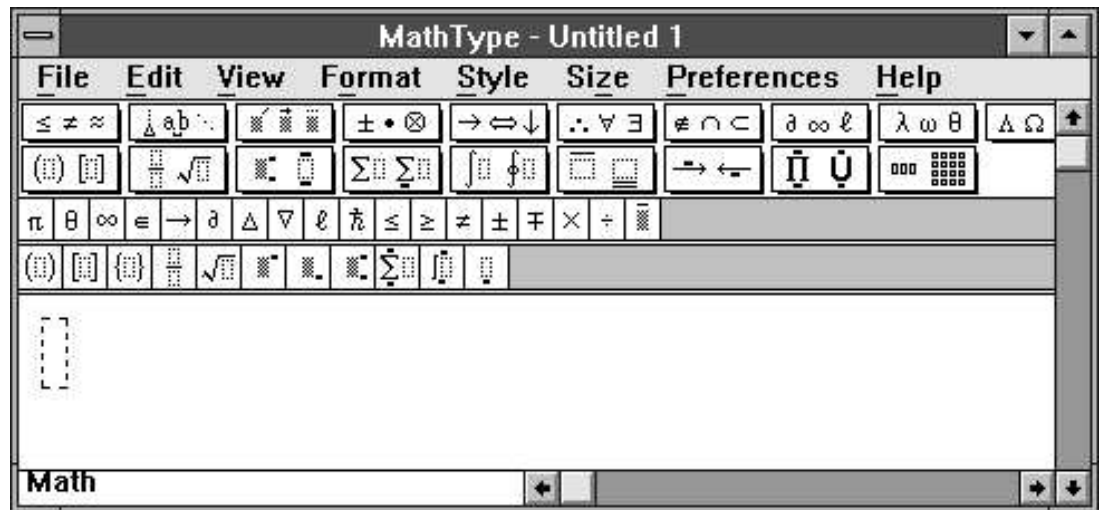
1. 화일메뉴의 새화일을 선택한다. 이때 서브메뉴에서 한글 논문 양식을 선택하면 논문 양식에 필요한 기본적인  $\text{LaTeX}$  화일의 골격이 입력된 윈도우가 화면에 생긴다.
2. 이때 %로 시작된 줄은 주석으로 그대로 두면 출력문서에 영향을 주지 않는다. 만일 예를 들어 페이지 크기, 위치 및 행간 지정을 하고자 하면 %를 지우고 필요하다면 숫자를 바꾼다.
3. 제목, 저자, 날짜 항목을 고쳐 원하는 내용을 입력한다.
4. 간단한 문서의 내용을 입력한다. 이때 `\section{ }` 명령어의 { } 내에도 입력해 본다.
5. 필요없는 `\section{ }` 혹은 `\subsection{ }` 명령어를 지운다.



6. 조판메뉴에서 LaTeX을 선택하거나 툴바에서 LaTeX 버튼을 누르면 저장될 화일의 이름을 물어보는 다이얼로그 박스가 나타난다.
7. 화일의 이름을 입력하고 OK 버튼을 누르면 DOS 박스(HLX)가 나타나면서 이 화일을 컴파일하게 된다.
8. 에러가 없이 컴파일이 끝나면 HLX 윈도우는 "비활성" HLX로 제목이 바뀐다.
9. 비활성 HLX 윈도우의 왼쪽상단에 있는 컨트롤 버튼을 마우스로 두번 클릭하면 이 윈도우가 닫혀진다.
10. 조판메뉴 혹은 툴바의 DVI 보기를 선택하면 DVI 보기 프로그램이 실행되어 컴파일된 화일의 내용이 나타난다. 이때 원한다면 프린터로 인쇄할 수도 있다.

TEX 혹은 LaTeX의 사용시 수식이 나오는 경우 굳이 수식에 관한 명령을 암기할 필요는 없다. 윈레이텍에 있는 MathType을 사용하여 WYSIWYG으로 수식을 입력하여 이에 해당되는 TEX 명령어를 얻으면 된다.

1. 화일메뉴의 새화일을 선택하여 한글 논문 양식을 선택한다.
2. 원하는 글의 내용을 입력한다.
3. 수식을 입력하고자 하면 툴바의 수식 에디터 버튼을 누르면 MathType이 구동되어 초기화면이 나타난다.



4. 원하는 수식을 만든다.
5. 수식의 입력이 끝났으면 마우스를 버튼을 누른채 수식전체를 끌어당기거나 Edit 메뉴의 Select All을 사용하여 수식전체를 선택한다. 이때 선택된 수식은 역상(reverse)으로 화면에 나타난다.

6. Preferences 메뉴의 Clipboard Format 을 선택하여 TEX Code<sup>1</sup> 를 선택한다. 이때 정해진 사항은 저장이 되므로 다음 수식의 경우에 다시 Preferences 메뉴로 들어와 정할 필요는 없다.
7. Edit 메뉴의 Copy 를 선택하여 선택된 수식이 클립보드에 복사되게 한다.
8. 윈레이텍으로 돌아와 수식이 들어갈 위치에 커서를 정하고 편집 메뉴의 붙이기를 사용하면 TEX 코드가 입력된다.
9. 이때 MathType의 내부 명령어가 주석부분으로 따라 입력되는데 만일 이 수식을 다시 MathType으로 갖고가 변경할 사유가 없는 경우에는 주석부분을 지워도 무방하다. 그러나 다시 이 수식을 변경할 경우가 예상되면 주석부분을 그대로 나둔다.
10. 화일을 컴파일하고 DVI 보기로 결과를 확인해 본다.

문서내에 그림을 넣고자하면 먼저 그림을 컴퓨터내에 원하는 형식으로 저장해야 한다. 보통은 WMF(Windows Meta File) 형식을 사용하고, 만일 출력 프린터가 포스트스크립트 프린터인 경우에는 EPS(Encapsulated PostScript) 형식을 준비한다. 이때 EPS로 준비된 그림은 DVI 보기 프로그램의 화면에 나타나지는 않지만 프린터에는 출력이 된다.

만일 원하는 그림이 윈도우즈의 클립보드내에 있다면 clipart를 사용하여 WMF 화일을 만들 수 있다. 또한 대부분의 그림 프로그램은 WMF 형식으로 export 하기 때문에 그림 프로그램의 선택은 다양할 수 있다.

---

<sup>1</sup> 텍스트 형식



## 4장

# 주프로그램 설명

주 프로그램은  $\text{TeX}$  파일 작성과  $\text{TeX}$  컴파일 명령의 수행 및 그 결과를 보기 위한 여러가지 기능을 갖춘 텍스트 에디터이다.  $\text{LaTeX}$  명령어의 입력을 도와주는 다양한 기능을 메뉴 및 다이얼로그 박스로 구현함으로써  $\text{TeX}$ 의 사용을 더욱 쉽게 하였다.

주 프로그램의 사용은 메뉴, 툴바의 버튼, 혹은 키보드의 입력을 통해서 이루어진다. 자주 사용하는 메뉴는 메뉴바 밑에 작은 그림 버튼으로 표시되어 있는데 이를 툴바라 한다. 마우스를 사용하지 않고  $\text{Ctrl}$  키와 같이 동시에 여러가지 키를 눌러 메뉴를 선택할 수 있는데 이를 단축키라 하며 메뉴 텍스트의 우측에 표시되어 있다.

마우스의 좌측 버튼을 텍스트 영역에 눌린 상태에서 움직이면 역상영역이 생기면서 텍스트의 일부분을 선택하게 된다. 이 선택영역을 편집 메뉴나 입력메뉴에서 여러가지 형태로 편집할 수 있다.

마우스의 좌측 버튼을 텍스트 영역에서 빠르게 두번 연속해서 누르면 그 지정된 단어를 선택하게 된다. 세번 연속해서 누르면 그 가르킨

줄이 선택되고, 네번이면 한 문단이 지정된다. 문단은 두개의 공백줄 (blank line)사이의 텍스트로 구분된다.

다음 설명에서 메뉴는 네모상자안에 이름이 표시되어 있고 그에 해당하는 툴바가 있는 경우 메뉴상자앞에 그 버튼 그림이 그려져 있다. 단축키는 둥근 상자 안에 표시되어있다.

## 4.1 파일메뉴



새화일

\$ \\_ \\_ \\_ \\_ !  
# Ctrl-N

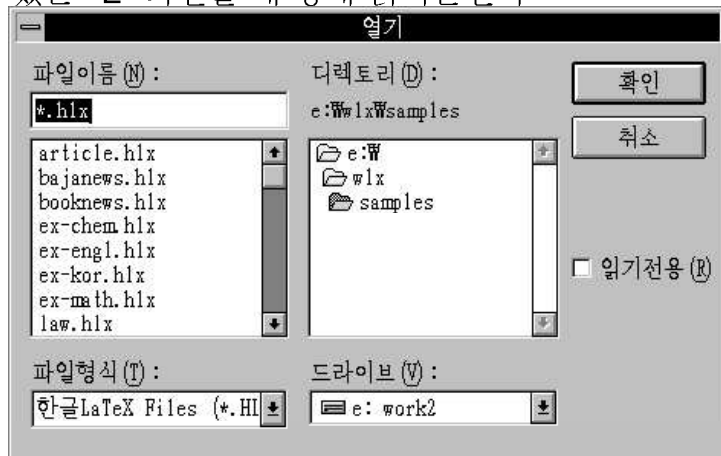
새로운 화일을 만들기 위한 창을 연다. 여러가지 기본양식의 이름들이 나열된 다이얼로그 박스가 나타나며, 특정양식을 선택하면 그 양식에 필요한 기본적인 TeX 화일의 골격이 미리 입력된 창이 열린다.



열기

\$ \\_ \\_ \\_ \\_ !  
# Ctrl-O

디스크에 있는 TeX 화일을 새 창에 읽어들인다.



저장

\$ \\_ \\_ \\_ \\_ !  
# Ctrl-S

현재 작업중인 TeX 화일을 저장한다.

저장(모두)

현재 열려 있는 모든 TeX 화일을 저장한다.



**저장(다른이름으로)**    작업중인 TeX 화일을 다른 이름으로 저장한다.

**저장 닫기**    현재 작업중인 TeX 화일을 저장하고 창을 닫는다.

**인쇄**    \$ \_\_\_\_\_!  
#Ctrl-P)    작업중인 TeX 화일을 인쇄한다(입력화일(ASCII)를 인쇄).

**HTML 입력**    .HTM 화일 확장자로 끝나는 HTML 화일을 읽어서 한글 LaTeX 화일로 바꾸어 읽어 들인다. 다음절(4.8)에서 자세히 설명되었다.

**HTML 출력**    한글 LaTeX 화일을 HTML 화일로 바꾸어 출력한다. 다음절(??)에서 자세히 설명되었다.

**인쇄(모두)**    현재 열려 있는 모든 TeX 입력소스화일(ASCII)을 인쇄한다.

**기타선택 설정**    인쇄종이, 프린터, WinTeX 윈도우 형식, 폰트등의 설정을 위한 서브메뉴가 나타난다.

**인쇄종이 형식 설정**

입력화일(ASCII)을 인쇄할 때 페이지 상단 및 하단에 인쇄할 내용을 지정할 수 있는 다이얼로그 박스가 나타난다.

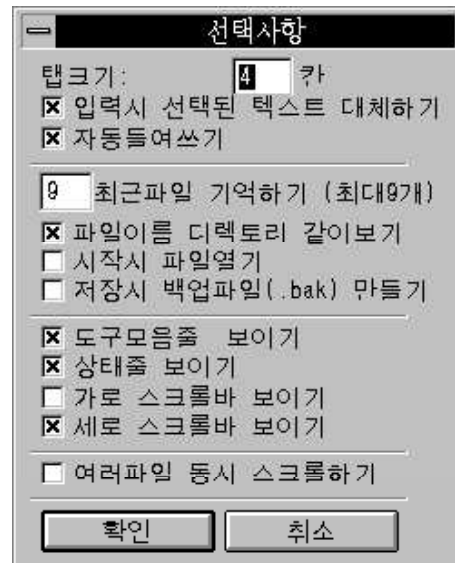


## 원레이텍 기본설정

원레이텍의 여러가지 선택사항을 지정할 수 있는 다이얼로그 박스가 나타난다.

탭 사이즈 지정, 줄 바꿀시 자동으로 **INDENT** 하기, 기억할 파일 이름의 갯수(최대 9개), 화일을 저장할때 **.BAK**의 확장자를 가진 백업화일 만들기등을 위한 선택을 할수있다.

주 프로그램의 툴바, 스크롤바등의 표시도 선택할수 있고, 두개이상 열린 화일을 동시에 스크롤하기위한 **Synchronize Text** 선택 사항도 있다.

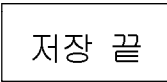


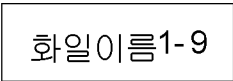
## 폰트 설정

인쇄할 때 및 화면상의 사용될 폰트를 선택할 수 있다.


끝


원레이텍의 수행을 끝낸다.


 저장 끝 현재 열려 있는 모든 TeX 파일을 저장하고 한글 WinTeX의 수행을 끝낸다.


 파일이름1-9  
파일 메뉴의 밑에는 최근에 열었던 파일들의 이름이 나열되어 있습니다. 원하는 파일의 이름을 마우스로 클릭하여 열 수 있습니다.


## 4.2 편집메뉴


 취소  $\$ \underline{\hspace{1cm}} !$   
 $\# \text{Ctrl-Q}$  입력 혹은 삭제등과 같은 실행을 취소하고 원상태로 복귀한다.

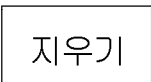
 다시 복귀되기 전 상태로 다시 돌아간다.

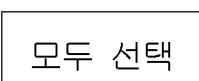
 반복 이전의 작업을 반복한다.

 자르기  $\$ \underline{\hspace{1cm}} !$   
 $\# \text{Ctrl-X}$  선택된 글을 잘라 클립보드에 저장한다.

 복사  $\$ \underline{\hspace{1cm}} !$   
 $\# \text{Ctrl-C}$  선택된 글을 복사하여 클립보드에 저장한다.

 붙이기  $\$ \underline{\hspace{1cm}} !$   
 $\# \text{Ctrl-V}$  자르거나 복사를 하여 클립보드에 저장된 부분을 현재 커서의 위치에 붙인다.

 지우기 선택된 글을 지운다.

 모두 선택 작업중인 파일의 내용을 전부 선택한다.

대문자로    선택된 글의 알파벳을 대문자로 바꾼다.

소문자로    선택된 글의 알파벳을 소문자로 바꾼다.

### 4.3    검색메뉴



찾기    \$ \_\_\_\_\_!  
#Qrl-F#

단어(들)을 입력할 수 있는 다이얼로그 박스가 열린다. 원하는 단어(들)을 입력하고 옵션들을 선택한 뒤 확인버튼을 누르면 그 단어(들)이 있는 부분으로 찾아간다.



계속 찾기    \$ \_\_\_\_\_!  
#Qrl-A# 현재 검색 중인 단어가 있는 다음 위치로 찾아간다.



바꾸기    \$ \_\_\_\_\_!  
#Qrl-R#

단어(들)을 입력할 수 있는 다이얼로그 박스가 열린다. 찾을 단어(들)와 그 단어(들)를 대치할 단어(들)를 입력하고 옵션들을 선택한 뒤 확인버튼을 누르면 그 단어(들)이 있는 부분으로 찾아간다. 단어를 바꾸어 넣을지를 묻는 다이얼로그 박스가 나온다.



- 줄로
 $\$ \text{Ctrl-G} !$  원하는 줄로 이동한다.
- 최근 바꾼곳으로 가장 최근 단어를 바꾼 지점으로 찾아간다.
- 표시 현재 선택된 부분에 표시를 한다.
- 표시한 곳으로 표시가 되어 있는 위치로 찾아간다.

## 4.4 입력메뉴

이 메뉴와 여기있는 두가지 다이얼로그 박스(LaTeX 명령입력 및 LaTeX 기호입력)를 사용하여 여러가지 LaTeX 명령어를 쉽게 입력할 수 있다. 이 중 글자모양, 글자크기, 문단 정렬, 문서 구조, 색인 등은 먼저 해당하는 부분을 선택한 뒤 명령어를 입력하여야 한다.

- 글자모양

선택된 부분의 글자모양을 지정하는 LaTeX 명령어를 삽입한다. 한글 글자모양에는 명조, **궁서**, 고딕, 둥근고딕, **진하게**, 경사, 밑줄치기 등이 있으며, 영문 글자모양에는 roman, bold, typewriter, italic, sans serif, slanted, small capital, underline 등이 있다.
- 글자크기

선택된 부분의 글자크기를 지정하는 LaTeX 명령어를 삽입한다. 글자크기에는 (크기순으로) tiny, scriptsize, footnotesize, small, normalsize, large, Large, LARGE, huge, Huge 등이 있다.
- 문단 정렬

선택된 부분의 문단 정렬을 위한 LaTeX 명령어를 삽입한다. 문단

정렬 방식에는 중앙 정렬, 좌측 정렬, 우측 정렬, 주석(footnote) 및 verbatim, quote, verse, table, parbox 등이 있다.

### 항목 정렬

항목 정렬을 위한  $\text{LaTeX}$  명령어를 삽입한다. 항목 정렬 방식에는 번호(enumerate), 점(itemize), 설명(description) 등이 있으며, 이 중 하나를 선택하면 항목의 수를 지정할 수 있는 다이얼로그 박스가 나타난다.

### 문서구조

선택된 부분에 특정 문서구조를 지정하기 위한  $\text{LaTeX}$  명령어를 삽입한다. 문서구조에는 부(part), 장(chapter), 절(section), 부절(subsection) 등이 있다.

### 색인

선택된 부분에 색인을 넣기 위한  $\text{LaTeX}$  명령어를 삽입한다. 여기에 해당하는 명령어는 label, ref, cite, index 등이 있다.

### LaTeX 기호입력

여러가지 기호에 해당하는  $\text{LaTeX}$  명령어를 버튼을 눌러 입력할 수 있는 다이얼로그 박스가 나타난다.



### 문자표

윈도우즈의 문자표프로그램(charmap.exe)이 수행된다. 이 프로그램을 사용하여 윈도우즈에서 쓸 수 있는 여러가지 특수문자들을

입력화일에 입력할 수 있다. 이 경우 입력되는 것은 특수문자 그 자체로서, 그 문자를 표현하기 위한  $\text{LaTeX}$  명령어가 아님에 주의한다.

**수식입력(MathType)**

한글 Win $\text{TeX}$ 에 있는 MathType 프로그램이 수행된다. MathType을 사용하면  $\text{TeX}$  혹은  $\text{LaTeX}$ 의 사용시 수식에 관한 명령을 암기할 필요가 없다. MathType을 사용하여 WYSIWYG으로 수식을 만들면 이에 해당되는  $\text{TeX}$  명령어를 MathType이 만들어 주게된다. 자세한 사용법은 MathType의 도움말을 참조할 것.

**LaTeX 명령입력**

$\$ \text{Ctrl-E} !$

여러가지  $\text{LaTeX}$  명령어를 버튼을 눌러 입력할 수 있는 다이얼로그박스가 나타난다.

이 툴 박스를 사용하여 항목 정렬을 할 경우 항목 수, -gure 명령어를 입력할 경우 파일 이름, tabular, tabbing, array 명령어를 입력할 경우 행 및 열 수 등을 지정하면 Win $\text{TeX}$ 이 자동으로 명령어들을 입력하므로 매우 편리하다.



**클립메타(Clipmeta)**

클립메타 프로그램이 수행된다. 클립메타 프로그램은 윈도우즈의 클립보드내에 있는 내용을 WMF 파일로 만들어 줌으로써 -gure

명령어를 사용하여 그림을 입력할 수 있도록 하여준다.

파일 읽어넣기

삽입할 파일을 선택할 수 있는 다이얼로그 박스가 나타난다. 이 다이얼로그 박스에서 파일을 선택하면 현재 커서의 위치에 선택된 파일 전체가 삽입된다.

## 4.5 조판메뉴



TeX 컴파일

\$  
#Orl-T!

현재의 입력화일을 컴파일한다. 입력화일의 형식에 따라 **TEX** 혹은 **LaTeX**이 자동으로 선택된다. 화일의 확장자가 **hl x**이면 **hl x2t ex**의 코드 변환과 함께 한글 **LaTeX**로 컴파일되며, **ht x**이면 **ht x2t ex** 코드 변환과 함께 한글 **TEX**로 컴파일되며, **t ex**이면 영어 **LaTeX 2.09**로서 컴파일된다. 먼저 현재의 화일이 저장되고 **DOS** 박스(**LATEX** 혹은 **TEX**)가 나타나면서 화일을 컴파일하게 된다. 에러가 없이 컴파일이 끝나면 **DOS** 박스 윈도우는 '비활성'으로 제목이 바뀌게 되며 윈도우의 왼쪽상단에 있는 컨트롤 버튼을 마우스로 더블클릭하면 윈도우가 닫힌다. 컴파일이 끝나고 나면 같은 화일이름에 확장자가 **.dvi**인 출력화일(**DVI** 화일)이 만들어진다.



DM 보기/인쇄

\$  
#Orl-D!

**DVIWIN** 프로그램이 실행되어 컴파일된 화일의 내용(실제 인쇄될 문서모양)이 나타난다. 한 페이지씩 원하는 페이지를 볼 수 있으며, 확대하거나 축소하여 볼 수도 있다. 자세한 **DVIWIN** 사용 방법은 **DVIWIN** 프로그램의 도움말을 참조할 것. 컴파일된 화일의 인쇄도 **DVIWIN**에서 할 수 있다.

LOG 화일 보기



TEX 컴파일 과정이 기록된 로그 파일을 볼 수 있다.

**BI BTEX**

도스박스안에서 `bibtex.exe`를 현재열려있는 파일이름으로 수행한다.

**MAKE INDEX**

도스박스안에서 `makeindx.exe`를 현재열려있는 파일이름으로 수행한다.

## 4.6 창메뉴

**연속정렬** 현재의 창들을 연속으로 겹쳐서 정렬한다.

**타일정일** 현재의 창들을 타일 형태로 정렬한다.

**모두닫기** 현재의 모든 창들을 닫는다.

**저장하고 모두닫기** 현재의 모든 파일들을 저장한 뒤 모든 창들을 닫는다.

**작업중인 창들**

창 메뉴의 밑에는 현재 작업중인 모든 창들의 이름이 나열되어 있다. 원하는 창의 이름을 마우스로 클릭하면 그 창으로 갈 수 있다.

## 4.7 도움말 메뉴

**목차** Help의 목차내용을 보여준다.

WnTex는?

## 4.8 HTML 파일에서 $\text{\LaTeX}$ 파일로 읽어들이기

본 변환기는 인터넷의 WWW 포맷의 문서를  $\text{\LaTeX}$  무서로 바꾸어 인쇄할 수 있도록 한다. HTML 1.2 사양을 기준으로 문서를 읽어들이며, ANCHOR, HEADER, LISTING, IMG 등의 요소와 여러 문자 Entity (ISO Latin 1)를 처리할 수 있다. 같은 파일내의 ANCHOR는 해당되는 페이지 번호로 연결시키었으나, 다른 파일일 경우에는 그 URL 이름을 주석란에 나타내도록 하였다.

$\text{\LaTeX}$ 의 특성상 FORMAT, 각 요소의 여러 특성변수 (ATTRIB) 는 처리하지 않았으나, 본문의 내용은 항상 유지되도록 하였다.

프로그램 실행은 Win $\text{\TeX}$  환경내에서 파일 메뉴중 HTML 입력 선택하면 파일 이름 입력 다이얼로그 박스가 나타난다. 입력할 HTML 파일 이름을 지정하면 변환하여 읽어들이는  $\text{\LaTeX}$  파일을 보여준다.

HTML 변환기는 한글 Win $\text{\TeX}$  환경하에서 뿐만 아니라, 일반적인 도스 코멘드 라인에서도 독립된 프로그램으로 다음과 같이 사용될 수 있다. 내부적으로는 h2latex.exe 라는 도스 프로그램을 이용하여 HTML 파일로 변환한다. h2latex.exe 는 h2x.pif 를 통하여 주 프로그램 (wintex.exe) 이 구동한다. 그 선택계수는 다음과 같다.

```
h2latex.exe [opt ...] [-le ...]
```

선택계수 opt 는 다음중의 하나 또는 그 이상이다.

```
-a author string
```

```
-c table of contents flag
```

```
-e outfile extension string
```

```
-f figure file appear flag (img의 입력파일 이름을 보여줌)
```

```
-n numbering section flag
```

```
-o  $\text{\LaTeX}$ \documentstyle option string
```

- p page break flag after title/table of contents
- s write to stdout flag
- t title string (제목)
- v Verbose flag (에러 및 변환상태 보고)

## 4.9 $\text{\LaTeX}$ 파일로 부터 HTML 파일로의 변환

WinTeX 프로그램에 포함 되어 있는 필터를 사용하면 기존의  $\text{\LaTeX}$  파일을 HTML 파일로 변환할 수 있다. WinTeX 환경내에서, 파일 메뉴중 EXPORT 를 선택하면 현재 열려져있는  $\text{\LaTeX}$  파일이 HTML 파일로 변환 된다. 이 절에서는 이 변환기의 기능과 변환시의 고려 사항들에 대해서 설명하기로 한다.

### 4.9.1 왜 HTML 로의 변환이 필요한가 ?

HTML 파일은 인터넷 환경하에서 온라인 정보를 저장하는 수단으로 가장 널리 쓰이고 있다. 따라서, HTML 파일 작성을 위한 여러 도구 들도 많이 소개되어 있는 상태이다. 그런데 만일 기존의 문서가  $\text{\LaTeX}$  용으로 만들어져 있다면, 이 때 이 변환기를 사용하여 바로 기존의  $\text{\LaTeX}$  문서를 온라인용 정보로 변환할 수 있게 된다. 온라인 문서로  $\text{\LaTeX}$  을 쓰는 또 다른 장점은, HTML 과 같은 하이퍼 텍스트 환경에서 필수적인 각 구성요소들 사이의 link 를 전혀 신경 쓰지 않아도, 이 변환기를 통하면 자동으로 각 구성요소들 사이의 link 가 설정 된다는 것이다.

이 밖에도, HTML 변환기는 문서를 일관적으로 관리하는데에도 큰 도움을 줄 수 있다. 예를 들어, 모든 문서는  $\text{\LaTeX}$  용으로 작성하여 관리 하되, 많일 필요시에는 언제든지 HTML 로 바꾸어 온라인 정보로 활용할 수 있게 된다. 그리고, HTML 자체가 하이퍼 텍스트 형태로 되어있기 때문에, 만일  $\text{\LaTeX}$  으로 작성 되어있는 매뉴얼이 있다면, 이를 온라인 도움말 형태로 바꾸는 것도 변환기를 이용하여 용이하게 할 수 있다.

앞서 간략히 언급한 바와 같이, HTML 의 중요한 요소는 관련된 자료들 사이의 연결 기능이다.  $\text{\LaTeX}$  문서는 일반적으로 여러개의 장(chapter), 절(section) 등 다양한 구조를 지니고 있는데, 이러한 구조들은 HTML 로

변환 되었을 때 link 형태로 바뀌어 서로 논리적으로 연결 되어 있는 부분을 용이하게 찾아 갈 수 있게 하여 준다. 그리고, 각주(footnote)등과 같이 본문 밖에 있는 내용들도, HTML의 link를 사용 하면 쉽게 그 내용을 참조 할 수 있다.

결론적으로 말하여, 이미 방대한 양의 문서들이, 특히 자연과학및 공학 계통의 전문 문서들이,  $\text{LaTeX}$ 을 이용하여 작성 되었으며, HTML 변환기를 사용하면 최단 시간내에 이를 온-라인화 하여 사용할 수 있게 된다.

#### 4.9.2 HTML 변환기의 기능

한글 WinTeX에 포함되어 있는 HTML 변환기의 기능을 열거하면 다음과 같다:

- 0 각 장 및 절 별로의 link 설정 |  $\text{LaTeX}$ 문서를 각 장 및 절 별로 분리하여 각각에 대한 link를 설정한다. 이 변환기를 통해 만들어진 HTML문서는 사용자의 옵션에 따라, 하나의 화일로 저장 되게 할 수도 있고, 또한 각각의 장/ 절 들이 모두 다를 화일에 저장 될 수도 있다. 기본 옵션은, 각각 다른 화일로 저장 하는 것인데, 네트워크 환경, 특히 모뎀을 통한 다이얼-업 통신을 고려한 시스템에서는, 가능한한 문서들을 작은 사이즈로 보관하는 것이 문서(HTML 문서) 전송시 발생하는 지연을 사용자로 하여금 적게 느끼게 할 수 있는 한 방법이기 때문이다.
- 0 문서의 포매팅 | HTML에서 가능한한 최대한으로 원래  $\text{LaTeX}$ 화일과 가깝게 포매팅을 한다. 이러한 포매팅은 HTML browser에 따라 다르게 보일 수 있는데, 이 프로그램에서는, 가장 널리 쓰이는 browser인 "netscape"의 포매팅을 주로 참조 하였다.  $\text{LaTeX}$ 의 list

환경 (`itemize`, `enumerate`, `quote` 등), 그리고 `\centerline`과 같은 포맷 지정 등, 대부분의 HTML browser에서 가능한 포맷 기능들을 포함한다. 또한 HTML 3의 table 기능을 사용하여, `\tabular` 환경도 HTML로 변환 가능하다. 하지만, 온-라인 browser에서 거의 무의미하다고 할 수 있는 쪽 설정 (page setup)에 관한 LaTeX 명령어(예를 들면, header나 footer, 그리고 용지 크기등을 지정하는 명령어)들은 모두 무시된다.

0 각주(footnote) 및 기타 보조 정보들과의 link | 이 부분은 아직 완성 되지는 않은 단계이나, 현재 모든 각주들이 해당되는 부분으로 link가 설정되어 있어서 바로 그 내용을 참조 할 수 있다. 앞으로, 각주 뿐만 아니라, index, 참고 문헌등도 동일한 방법으로 참조 할 수 있는 기능을 갖추게 될 것이다.

0 사용자가 정의한 매크로 및 environment들의 처리 | LaTeX을 사용해본 사람이라면, 누구나 자기 자신의 매크로를 정의하여 사용한 경험이 있을 것이다. 따라서, 기본적인 LaTeX 명령어 이외에 이와 같이 사용자가 `\newcommand`나 `\newenvironment` 같은 명령어를 사용하여 새로이 정의된 명령어 또는 환경을 완벽하게 처리할 수 있도록 하였다. 한 가지 염두에 둘것은, 본 변환기는, LaTeX문서를 가정하였으므로 보다 하위레벨의 TeX 명령어들은 대부분 처리 할 수 없다는 것이다. LaTeX을 사용 하더라도, TeX 명령어를 빈번히 쓴 경우에는 그 변환 결과가 흡족할 수 없다는 것을 이해해야 할 것이다. 따라서, 이 변환기에서는 자주 사용되는 custom 마크로 패키지들을 직접 지원하도록 하고 있다. 예를 들면 `\psfig`, `\code` 그리고 `\epsf` 등 많이 사용되는 패키지의 명령어들을 직접 이 변환기가 처리한다.

- 0 그래픽 화일의 입력기능 | JPEG 화일 (.jpg) GIF 화일 (.gif) 들을 HTML 문서에서 상용할 수 있도록 하는데, 자세한 것은 뒤에서 설명하도록 한다.
- 0 TEXINPUTS 환경변수의 이용 | LaTeX에서 일반적으로 사용되는 이 환경변수를 인식하며, 이 변수에 지정된 디렉토리들로부터 화일을 자동으로 검색한다. 단, 스타일(\*.sty) 화일들은 무시한다. 이는, TeX 명령어를 처리하는데 따르는 제약 때문이다.
- 0 MS-DOS 및 모든 UNIX 환경에서 동일하게 작동 | 이 변환기는, MS-DOS에서 뿐만 아니라, 윈도우스 (DOS-BOX 하에서) 그리고 여러 UNIX 시스템에 설치되어 시험되었으며, 모두 동일 하게 작동을 한다. 또한 한글이나 영문 및 기타 다중 바이트 문자들을 처리 할 수 있다.

이상 HTML 변환기의 기능을 간략히 소개 하였는데, 물론 LaTeX이 가지고 있는 수많은 포매팅 기능을 HTML에서 모두 소화 할 수는 없다. 특히, LaTeX의 가장 중요한 기능인 수학 공식이 현재의 HTML에서는 불가능 하다. 하지만, 앞으로의 HTML 표준으로 기대되는 HTML 3가 보편화 되면, 이에 맞게 수학기호나 기호 그리고 복잡한 도표들도 LaTeX과 비슷한 수준으로 포매팅 할 수 있을 것으로 예상된다. 이에 따라, 이 변환기의 기능도 더욱 향상 될 수 있을 것이다.

### 4.9.3 도스 코맨드 라인에서 변환기의 수행

HTML 변환기는 WinTeX 환경하에서 뿐만 아니라, 일반적인 도스 코맨드 라인에서도 독립된 프로그램으로 다음과 같이 사용될 수 있다.

```
t ex2x options... | at ex_file_name
```



이를 보다 자세히 열거하면:

`l a t e x _ f i l e _ n a m e`

입력되는  $\text{LaTeX}$  화일. 만일 여러 화일이 포함된 경우에는 루트 화일 지정.

또한 각각의 옵션으로는:

`- o u t p u t _ d i r e c t o r y`

생성된 **HTML** 화일이 저장되는 디렉토리. 하나의  $\text{LaTeX}$  화일로 부터, 적게는 한개, 많게는 수백개의 **HTML** 화일이 생성될 수 있는데, 이 화일들이 저장될 디렉토리를 지정한다. 이 옵션이 생략될 경우, 기본 설정으로, `\HTML_DIR` 이란 디렉토리가 현재 디렉토리 아래에 만들어 진다.

`- p e r r o r`

**HTML** 변환기에서 발견된  $\text{LaTeX}$  화일의 문법적인 오류나 또는 지원이 안되는 명령어들을 출력한다. 하지만  $\text{LaTeX}$  화일의 오류가 발견 되어도, 대부분의 경우 **HTML** 화일을 생성하는데는 지장이 없다. 왜냐하면, 이러한 오류들이 발견되면, 그 내용이 **HTML** 화일에 포함되기 때문에 나중에 얼마든지 이러한 오류들을 추적할 수 있다.

`- n o s p l i t`

한개의 **HTML** 화일로 만들어 지도록 한다. 앞서 언급되었듯이, 기본적으로는,  $\text{LaTeX}$  화일의 구조에 따라 여러 **HTML** 화일들이 생성되나, 이 옵션을 통해 한개의 화일로 만들 수 있다.

-gdi r dir\_name

그래픽 파일들이 저장되는 디렉토리를 지정한다. 이 디렉토리는, HTML 디렉토리 아래에 생기며, 모든 그래픽 파일들 (JPEG, GIF 파일)이 이 디렉토리에 저장 된다. 이 옵션이 생략 될 경우 `\GRAPHICS` 라는 디렉토리가 HTML 디렉토리 아래에 만들어 진다.

#### 4.9.4 테이블 및 그래픽스의 사용

HTML 변환기를 사용하는데 있어서 고려 되어야 할 사항, 특히, 테이블과 그래픽스 사용에 관해 좀 더 살펴 보자.

##### HTML 테이블에 관해

HTML 3에서 테이블의 사용에 관한 내용을 정의 하고 있는데, 최근의 몇몇 browser에서 (netscape을 위시한) 이미 이 기능을 지원하기 시작하여, 본 HTML 변환기에도 `LaTeX`의 `\table` 환경을 HTML 3의 테이블 환경으로 변환하는 기능을 추가하였다. 하지만 `LaTeX`의 모든 테이블 기능을 살리는 것은 불가능하기 때문에, 여러 포맷 상의 차이를 감수해야 하는데 이는 다음과 같다:

- 0 우선 테이블의 외곽선 정의 방식에서 `LaTeX`과 HTML이 판이하게 다르다. `LaTeX`에서는 각 행 또는 열 사이에 임의로 외곽선을 집어 넣거나 빼 수 있지만, HTML에서는 표 전체에 외곽선을 그리거나 또는 아예 넣지 않거나 하는 두가지 옵션만 있다. 물론, HTML 3에서 새로이 정의한 스타일 쉬트를 사용하여, 좀 더 개성있는 테이블을 만들 수 있을 지 모르나, 결국은 HTML 브라우저에서 지원이 되지 않으면 별반 효과가 없게 된다.

0  $\LaTeX$ 에서는 단순한 선 대신, 일반 텍스트를 각 셀사이에 집어 넣을 수가 있는데 ( $\@xxxx$ )와 같은 변수를 사용하여, HTML에서는 불가능 하다.

0  $\backslash\text{cline}\{m:n\}$ "을 사용한 부분 선 그리기는 무시된다.

이상과 같이 몇가지 포맷 상의 제약은 있지만, 이 변환기를 통해 만들어진 테이블은 원래의 테이블과 동일한 행과 열 구조를 가지고 있으므로 ( $\backslash\text{multicol}$ 도 지원됨), 도표를 이해하는 데에는 큰 문제가 없으리라 본다. 또한  $\LaTeX$ 에서와 똑 같이 중첩된 테이블(테이블 셀안에 다른 테이블이 정의됨)도 HTML 화일로 변환이 가능하다.

### 그래픽스 화일의 사용에 관해

$\LaTeX$ 에서는 여러가지의 명령어를 사용하여 외부 그래픽스 화일을 포함할 수 있도록 한다.  $\LaTeX$ 에서 가장 보편적으로 사용되는 포맷은 역시  $\text{postscript}^{\text{TM}}$ (특히  $\text{encapsulated postscript}$ )인데, 이는 일반적으로  $\backslash\text{special}$ "이라는  $\text{TEX}$ 명령어를 이용해 구현된다. 하지만, 이  $\backslash\text{special}$ " 명령어를 사용할 때 일정한 규칙이 없으므로, 여러 패키지마다 그 사용 방법이 달라서, HTML 변환기가 모든 형태를 처리할 수는 없다. 따라서 다음과 같은 포맷으로 사용된  $\backslash\text{special}$ " 명령에 대해서만 처리할 수 있다.

```
\special{ action graphics_-le_name and whichever comes afterwards }
```

또는

```
\special{ action=graphics_-le_name and whichever comes afterwards }
```

예를 들면:

```
\special {i socal e pi c1. wrf 2i n 3i n}
```

```
\special {psfile=pic2. ps, width=2i n, height=3i n}
```

위의 예에서 볼 수 있듯이, HTML 변환기에서 그림의 크기는 무시한다. 그리고, 이 `\special` 명령어가 그래픽스 화일을 지정하는 것이 아닌 다른 용도로 쓰일 경우는 그냥 무시되어 버린다.

앞에서 이미 언급 하였듯이, 현재, HTML 변환기에서는 JPEG 화일과 GIF 화일만 HTML 문서에서 쓸 수 있도록 하였다. 이는 대부분의 browser가 이 두가지 화일에 대해서만 인라인 그래픽스를 허용하기 때문이다. 따라서, 위와 같이, JPEG 또는 GIF 화일이 아닌 포맷을 지정한 경우에, HTML 변환기는 상응되는 JPEG 화일 또는 GIF 화일이 동일한 디렉토리에 있다고 가정한다. 이를 테면, 위의 예에서와 같이 `\pic1.wrf` 화일을  $\text{\LaTeX}$  문서에서 지정한 경우, HTML 변환기는 `\pic1.jpg` 또는 `\pic1.gif` 화일이 같은 디렉토리에 있는것이 확인 될 경우에, 이 JPEG 혹은 GIF 화일을 GRAPHICS 디렉토리에 복사해 놓는다. 만일 해당되는 JPEG 또는 GIF 화일이 없을 경우라도, HTML 화일은 이 그래픽 화일들이 저장 되어야 할 곳을 가리키고 있기 때문에, 나중에라도 사용자가 해당되는 화일을 복사에 놓으면 된다. 그렇지 않으면, HTML browser를 사용 했을 때, 그림 대신 공란으로 이 부분이 표시 될 것이다. 결론적으로 말하여, HTML 변환기가 모든 형태의 그래픽스 화일을 처리할 수 없으므로, 필요한 JPEG/GIF 화일은 사용자가 준비하도록 한 것이다.

이 밖에 (`\special` 이외에), 일반적으로 많이 쓰는 다음과 같은 postscript 용 명령어를, HTML 변환기가 직접 처리 한다.

```
0 \psfig{ }에 의한 그림 입력.
```

0 `\epsffile{}`에 의한 그림 입력.

0 `\epsfbox{}`에 의한 그림 입력.

그림의 크기, 위치 조정과 관련된 옵션들은 HTML 파일 생성시 무시된다.

#### 4.9.5 앞으로의 버전에 대한 계획

현재의 HTML 변환기도 이미 상당히 많은 기능을 제공하고 있지만, 다음과 같은 미비점들을 계속 보완해 나갈 계획이다.

- 0 그래픽 파일들의 자동 변환 | GIF 파일 또는 JPEG 파일이 아닌 경우 (예를 들어 `postscript` 파일) 이를 자동으로 변환.
- 0 수학기호의 사용 | 이는 확실한 HTML 3 browser가 나온다면, 쉽게 구현될 수 있을 것이다.
- 0 보다 완벽한 크로스 레퍼런스 기능 | 현재 상태에서도 일부 구현되어 있기는 하나, 현재 지원이 되고 있지 않는 인덱스 기능과 참고문헌등을 지원할 것이다.



## 5장

# DVIWIN 설명

### 5.1 DVIWIN 이란?

DVIWIN은 MS-윈도우즈상에서 DVI 화일을 볼수있게 하는 프로그램이다. DVIWIN의 기본 개념은 현재의 페이지 내용을 비트맵으로 만들어 윈도우에 나타내는 것이다. 글자체는 T<sub>E</sub>X의 메타폰트에서 유래된 PK 폰트나 GF 폰트화일에서 읽어오거나, Windows에 설치된 TrueType 폰트에서 읽어들이 수 있다.

스크롤바나 메뉴, 혹은 단축키를 사용하여 문서의 어느 곳으로나 갈 수 있으며, 비트맵을 위한 프린터 드라이버와 충분한 기억용량만 있으면 윈도우에서 사용가능한 어느 프린터로나 문서를 출력할 수 있다. 또한 DVIWIN은 그림을 문서에 삽입하기 위하여 사용되는 T<sub>E</sub>X `\special {}` 명령어를 지원한다. DVIWIN은 윈도우즈 3.1상에서 동작

한다.<sup>1</sup>

## 5.2 화면 움직이기

문서의 한 페이지 위에서 윈도우의 위치를 일차계수, 이차계수, 혹은 한 pixel만큼 이동할 수 있으며, 양쪽 끝단이나 구석으로 바로 이동할 수도 있습니다. 일차계수 및 이차계수는 선택 메뉴에서 조정할 수 있다. 화면 이동을 위한 마우스 버튼 및 키 사용법은 다음과 같다.

어느 위치에서나 마우스의 왼쪽 버튼을 두번 누르면 그 지점이 왼쪽 위 구석에 오도록 윈도우가 움직이며, 오른쪽 버튼을 두번 누르면 그 지점이 오른쪽 아래 구석에 오도록 윈도우가 움직인다. 눈금자 위에서 버튼을 두번 누르면 윈도우는 그 자 방향으로 움직인다. 마지막으로, `|Space|` 키는 일차계수만큼, 그리고 `Enter` 키는 이차계수만큼 윈도우를 앞으로 이동시킨다. `|Ctrl-Space|` 키와 `|#Ctrl-Enter|` 키는 같은 만큼 윈도우를 뒤로 이동시킨다. 이 방법을 사용하면 하나의 키로 문서의 처음에서 끝까지 다 갈 수 있다.

## 5.3 그림 넣기

TEX에 그림 파일을 넣기 위하여 `\special{}` 명령어를 쓸 수 있다. 그림 파일을 읽어 들이는 방법과, `\special{}` 명령어를 쓰는 방법은 다음과 같다.

그림 파일을 읽어 들이는 방법에는 두가지가 있다. 첫번째 방법은 Windows의 메타 파일(wmf)을 이용하는 것이다. 이 방법은 빠르고

---

<sup>1</sup>DVIWIN은 가능한 많은 폰트 파일을 사용하므로 CONFIG.SYS 파일의 FILES= 문을 사용하여 최대 파일 수를 50이상으로 지정하도록 해야 한다.



간단하며, 여러가지 해상도로 바꿀 수 있고, 거의 모든 윈도우즈 응용 프로그램에서 사용될 수 있다는 장점이 있다. 이 방법의 한 가지 문제점은 응용 프로그램들이 화일을 메타 화일 형식으로 저장하지는 않는다는 것이다. 그러나 화일을 클립보드로 옮길 때는 이 형식을 씬으로, 그림을 클립보드로 옮긴 다음 메타 화일로 저장하여 사용할 수 있다. `clipmat a. exe`를 사용하면 이 작업을 쉽게 할 수 있다. 이 방법을 이용하여 Excel, PowerPoint, Word for Windows, 1-2-3 for Windows, Toolbook, Paintbrush 등으로 부터 그림을 읽어 들일 수 있다. 두번째 방법은 특정한 그림 화일을 읽어 들이는 DLL(Dynamic Link Library) 을 사용하는 것이다. 이들을 그래픽스 필터 라고 하는데, 여러가지 필터를 씬으로써 메타 화일을 사용하지 않고도 그림을 읽어 들일 수 있다. DVIWIN에는 PCX, BMP, 그리고 MSP 필터가 있으며, 그 외 다른 필터들도 사용할 수 있다. 특정한 화일에 사용할 필터들을 `win.ini` 화일의 [MS Graphic Import Files] 란에 `description=driver, extension` 형식으로 지정할 수 있다. 여기서 `description`은 특정 형식을 설명하는 문구 이며, `driver`는 필터의 이름과 경로, 그리고 `extension`은 그 형식의 화일에 붙이는 확장자이다. 따라서 DVIWIN의 필터들은 `win.ini` 화일에 다음과 같이 지정하면 된다:

```
[MS Graphic Import Filters]
PC Paint brush(. PCX) =XXXXX\pcxi n. flt, PCX
Microsoft Paint(. MSP) =XXXXX\mspi n. flt, MSP
Bitmaps(. BMP) =XXXXX\bmpi n. flt, BMP
Bitmaps(. DIB) =XXXXX\bmpi n. flt, DIB
Bitmaps(. RLE) =XXXXX\bmpi n. flt, RLE
```

XXXXX는 경로명을 나타낸다.

그림을 넣기위한 특수명령어 형식은 다음과 같다:

```
\special {action filename, x-paper-size y-paper-size}
```

-**lename** 은 읽고자 하는 그림 화일의 이름이다. 화일이 다른 경로에 있을 때에는 전체 경로명을 써 주어야 하며, 이 때 **n** 대신에 반드시 / 를 사용하여야 한다. **x-paper-size** 와 **y-paper-size** 는 원하는 그림의 크기를 나타내며, 숫자와 단위로 표현한다. 다음과 같은 단위들을 쓸 수 있다:

단위	이름	변환
in	inch	1in = 2.54cm
bp	big point	72bp = 1in
pt	point	72.27pt = 1in
pc	pica	1pc = 12pt
sp	scaled point	65536sp = 1pt
dd	didot point	1157dd = 1238pt
cc	cicero	1cc = 12dd
cm	centimeter	2.54cm = 1in
mm	millimeter	10mm = 1cm
px	pixel	

**action** 은 그림 화일을 어떻게 넣을 것인 지를 지정한다. **center** 는 그림을 중앙에 들어가게 하며, **isoscale**은 그림을 지정된 크기에 맞도록 **X-Y**에 대하여 같은 비율로 줄이고, **anisoscale**은 그림을 본래 **X-Y** 비율에 관계없이 지정된 크기로 줄인다. 예를 들어 **-gure.wmf**라는 메타 화일을 본래 **X-Y** 비율에 관계없이 너비 4.5cm, 높이 2.5cm로 넣고 싶다면 다음과 같이 하면 된다:

```
\begin{figure}[ht]  
\caption{XXX}  
\special {anisocal e figure.wmf , 4.5cm 2.5cm}
```

```
\vspace{2.5cm}
```

```
\end{figure}
```

참고로, 표준 메타 화일은 그림의 크기에 관한 정보를 가지고 있지 않으므로 `clipmeta.exe`를 사용할 때에는 항상 `placeable` 메타 화일을 쓰는 것이 바람직 하다. 이미 지정된 그림 크기를 바꾸고 싶을 때는 다음과 같이 할 수 있다:

```
\special{action filename,
```

```
x-paper-size y-paper-size x-figure-size y-figure-size}
```

`x--g-size` 와 `y--g-size` 는 그림의 크기를 다시 지정한다. 이 방법은 특히 비트맵으로 표현된 그림을 찌그러짐 없이 나타내는 데 유용하다. 예를 들어, 900#600 비트맵을 300dpi 프린터에 찍는다면, 비트맵의 크기는  $900/300 = 3\text{in}$  #  $600/300 = 2\text{in}$  가 될 것이다.

## 5.4 인쇄하기

프린트 다이얼로그를 이용하여 프린트할 페이지와 원하는 해상도를 선택할 수 있으며, 해상도에서 *자동*을 선택하면 **DVIWIN**이 지정된 프린터의 해상도에 가장 가까운 값을 자동으로 선택하게 되어 있다. *설치버튼*을 누르면 여러가지 프린터를 선택할 수 있는 또다른 다이얼로그로 갈 수 있다.

선택사항 중에는 *특수명령어 보냄*이 있다. 이것이 선택되어 있지 않으면 **DVIWIN**은 모든 특수명령어를 비트맵으로 만들어 전체 비트맵을 프린터로 보내고, 이것이 선택되어 있으면 특수명령어를 직접 프린터로 보낸다. 특수명령어를 직접 프린터로 보내면 프린터가 가지고 있는 특수 기능들을 쓸 수 있다는 장점이 있다. 예를 들면, 컬러 프린트나,

halftoning, dithering 등을 할 수 있다. 그러나 이 방법은 속도가 조금 느리며, 정확한 WYSIWYG이 아닐 수가 있다는 단점이 있다.

프린터에 따라 특수한 DVI 드라이버를 사용하고 싶은 경우, 프린트 다이얼로그의 외부 프린트에 DVIWIN이 프린트할 때 실행되는 명령어를 지정할 수 있다. 명령어가 실행되기 전에 지정되는 몇가지 특수값들은 다음과 같다.

- \$1            프린트할 첫번째 페이지
- \$2            프린트할 마지막 페이지
- \$3            페이지 간격.  
(연속적으로 모든 페이지를 프린트할 경우 = 1,  
두 페이지당 하나씩 프린트 할 경우 =2)
- \$r            수평 해상도
- \$x            수평 해상도
- \$y            수직 해상도
- \$X            페이지 폭 (i nches)
- \$Y            페이지 높이 (i nches)
- \$Rxxx        페이지가 역순일 경우 "xxx" 를 붙임
- \$b            dvi 파일 이름
- \$(xxx)        환경 변수 "xxx" 의 내용

예를 들어, dvips를 수행하고, 프린터에 PS 화일을 보낸다음 그 PS 화일을 지우고 싶으면, 다음과 같이 하면 된다.

1. rundvi ps. bat 화일을 다음과 같이 만든다.  

```
dvi ps -p=%1 -l=%2 %4 %8
copy %8. ps prn / b
del %8. ps
```

2. `rundvi ps. pi f` 파일을 만든다.
3. "외부 프린트"에 다음과 같이 지정한다.  
`rundvi ps. pi f $1 $2 $b $R-r`

다이얼로그 이외에, 외부 프린트를 가능하게 하는 체크박스과 외부 명령어를 편집할 수 있는 체크박스가 있습니다.

## 5.5 T<sub>E</sub>X 및 기타 프로그램과의 관계

DVIWIN은 일반적인 *편집 컴파일 보기* 과정을 단순화 하였다. 입력 초점(input focus)이 DVIWIN에서 떠날때 마다 DVIWIN은 현재의 DVI화일을 닫는다. 입력 초점이 다시 DVIWIN으로 오면, DVIWIN은 DVI화일에 변화가 있는 지를 보고, 변화가 있으면 화일을 새로 읽어 들인다. 이렇게 함으로써 T<sub>E</sub>X나 editor로부터 DVIWIN으로의 전환을 보다 쉽게 하였다. DVIWIN은 또한 다른 프로그램에서도 다음과 같이 부를 수 있다:

```
dvi win [-1] [-c] [-g NN] [-r] [filename]
```

-1은 항상 하나의 DVIWIN만이 수행되도록 한다. 만일 또다른 DVIWIN이 수행 중이면 앞의 DVIWIN은 중단된다. -c는 -1과 같이 쓰여, DVIWIN이 스스로 수행을 중단 할 수 있도록 한다. -g NN은 DVIWIN이 페이지 NN으로 가도록 하며, -r은 모든 폰트를 다시 읽어 들이도록 한다.

DVIWIN은 비트맵을 프린터 관리자(print manager)로 보냄으로써 문서를 출력하며, 스크린이나 프린터로의 출력을 위한 윈도우즈 드라이버에 의존하도록 되어있다. 비트맵의 크기가 64KByte를 넘는 경우, 문제를 일으키는 비디오 드라이버가 있을 수도 있다. 화면 출력에 문제

가 있는 경우에는 비디오 드라이버를 점검해야 한다. 현재 DVIWIN은 4MByte 램(RAM)이 있는 경우 최대 300dpi 까지 swapping 없이 동작하도록 되어있다. 만일 600dpi 이상을 사용하고자 할 경우는 6MByte 이상의 램이 있는것이 바람직하다.

## 5.6 메뉴 설명

파일 메뉴  $\$ \underline{\hspace{1cm}} ! \$ \underline{\hspace{1cm}} ! \$ \underline{\hspace{1cm}} !$   
 $\# \text{Ctrl-O} \# \text{Ctrl-C} \# \text{Ctrl-1-4}$

파일 메뉴에서는 파일을 불러오고, 닫고, 인쇄하고, 프린터를 지정하고, 수행을 끝낼 수 있다. 메뉴의 끝에는 최근에 보았던 4개의 파일 목록이 있다. 어느 파일이나 마우스를 클릭함으로써 열 수 있다. **프린트** 메뉴를 선택하면 여러가지 프린트 방식을 지정할 수 있는 다이얼로그 박스가 열린다. **끝(선택사항 저장)** 을 이용하여 수행을 끝내면 모든 변수값들이 저장되고, **끝** 을 이용하면 바뀐 값들이 저장되지 않고 수행이 끝난다.

**끝(선택사항 저장)**

페이지 메뉴  $\$ \underline{\hspace{1cm}} !$   
 $\# \text{Ctrl-G}$

페이지 메뉴를 이용하면 현재 페이지위의 다른 곳으로 가거나 다른 페이지로 갈 수 있다. **다른 페이지** 를 선택하면 보고자하는 페이지를 지정할 수 있다. 여기서 음수는 문서의 끝에서 부터의 상대적인 위치를 나타낸다. 즉, -1은 마지막 페이지를 지정하게 된다. 다음 페이지나 앞 페이지로 이동할 때는 수직 스크롤바에서 마우스를 클릭하거나, **PgUp** & **PgDn** 를 사용할 수도 있다. 우측상단에 **→** 또는 **←** 의 툴바 버튼도 사용할 수 있다. 아래쪽의 네가지 명령어를 쓰면 현재 페이지의 구석으로 이동할 수

있다. 이 명령어들을 한 번 누르면 문서 내용의 끝으로 가고, 한 번 더 누르면 실제 종이의 끝으로 가게 된다.



프린트는 현재 보고있는 dvi 화일을 프린터로 출력하기 위한 다이아로그 박스를 연다.

### 해상도 메뉴

해상도 메뉴에서는 문서의 해상도를 선택할 수 있다. 끝의 두 메뉴는 비어있으며, 선택메뉴의 사용자 정의 해상도를 이용하여 어떤 해상도든지 정해 넣을 수 있다.

### 확대 메뉴

확대 메뉴를 이용하면 페이지(전체)를 확대하거나 축소할 수 있다. 확대 혹은 축소비율은  $\left| + \left| \text{나} \right| \right|$  키를 눌러서 늘이거나 줄일 수도 있다. 우측상단에 또는 의 툴바 버튼도 사용할 수 있다.

### 선택 메뉴

선택 메뉴를 이용하면 DVIWIN의 동작과 관련된 몇가지 변수들을 지정할 수 있다. 모든 변수들은 dviwin.ini 화일에 저장된다.

- 폰트 추적 (TRACE)
- 자 보기 (RULERS)
- 수평 위치 고정 (KEEPHOR)
- 수직 위치 고정 (KEEPVER)
- 경고음 (BEEPS)
- 변수 저장S (AVEDEF)

흑백 (BLACK/WHITE)  
 폰트 새로 읽기 (RESET FONTS)  
 초기 위치 (MARGINS)  
 초기 위치로 선택 (APPLY MARGINS)  
 로그 파일 보기 (LOG MESSAGE)  
 사용자 지정 해상도 (CUSTOM SIZE)  
 폰트 캐쉬 (FONT CACHE)  
 스크롤 값 (SCROLL FACTOR)  
 돋보기 (MAGNIFY)  
 종이 크기 (PAGE SIZE)  
 색상 조절 (COLOR)

폰트 추적  $\$ \underline{\text{Ctrl-T}}$

폰트 추적이 선택되면 DVIWIN은 log 파일을 만들고 모든 폰트 관련 활동을 기록한다. 프로그램 수행중에 폰트에 관련된 문제가 생기면 dvi 파일을 읽기 전에 폰트 추적을 선택하고 파일을 읽어들이고 후 log 파일을 본다. DVIWIN은 한번에 최대 17개까지 폰트 파일을 사용한다. 만일 폰트가 있고, 올바른 경로가 주어졌는 데도 프로그램이 폰트를 찾지 못하면, con-g.sys 파일에 FILES=50 (혹은 그 이상) 이 지정되어 있는지 확인한다.

표시자  $\$ \underline{\text{Ctrl-U}}$

자 보기가 선택되면 커서는 십자로 바뀌고, 화면에 두개의 자 가 나타나며, 현재의 커서 위치가 자에 나타납니다. 문서의 위치에 대해 더 정확한 정보가 필요한 경우에는 왼쪽 마우스 버튼을 control 키와 함께 누른다. DVIWIN이 정확한 위치를



알려준다.

수평 유지

$\$ \overline{\text{Orl-H}}!$

수평 위치 고정이 선택되면 다른 페이지로 이동할 때 수평 위치가 바뀌지 않는다.

수직 유지

$\$ \overline{\text{Orl-V}}!$

수직 위치 고정이 선택되면 다른 페이지로 이동할 때 수직 위치가 바뀌지 않는다.

경고음

경고음이 선택되면, 사용자가 잘못된 명령어를 쓰면 경고음이 난다.

자동환경 저장

변수 저장이 선택되면 프로그램 수행이 끝날 때 변수들이 자동으로 저장된다.

흑백

$\$ \overline{\text{Orl-W}}!$


흑백이 선택되면, 화면 출력은 흰 바탕에 검은 글자로 나타난다.

PK 폰트쓰기

TrueType 폰트를 찾지 못하면 PK 폰트화일을 쓰도록 환경을 바꾼다.

로그 화일 보기

이 명령어를 쓰면 본 메뉴에서 *로그 화일* 명령어를 수행시킬 때 사용할 프로그램을 지정할 수 있다. `wbr.exe`를 비롯한 어떤 브라우저나 에디터도 쓸 수 있다. `wbr.exe`는 간단한 브라우저로서, 예를 들어 항상 `dvi` 로그 화일과 `(.lg)` TeX 로그 화일을 `(.log)` 모두 보려면 로그 화일 보기 명령을 다음과 같이

지정하면 된다. 우측상단에  의 툴바 버튼도 눌러 볼 수도 있고 만약 dviwin에서 에러가 발생시에는 이 버튼 색깔이 노란색이되고 치명적 에러에는 빨간색이 된다.

```
wbr -l $l $b.l og
```

\$l은 로그 파일 이름을 나타내는 변수이며 \$b는 dvi 파일 이름을 나타내는 변수이다.

#### 해상도지정

이 명령어를 사용하여 임의의 해상도를 2개 까지 지정할 수 있다. 수직및 수평 해상도를 모두 지정할 수 있으며, 수평 해상도를 0으로 하면 지정된 해상도를 취소할 수 있다.

#### 폰트 메모리

이 명령어를 사용하여 폰트를 읽어 들이는 메모리의 크기를 0에서 50까지 조절할 수 있다. 대부분의 문서는 메모리 크기가 20이면 충분하다.

#### 화면 움직이기 지정

이 명령어를 사용하여 스크롤바에서 마우스를 클릭하거나 화살표 키를 누를 때 윈도우가 움직이는 정도를 지정할 수 있다. 일차 계수는 마우스를 스크롤바에서 클릭할 때 (혹은 화살표 키에) 적용되는 값이며, 이차 계수는 마우스를 스크롤바 끝에 있는 화살표에서 클릭할 때 (혹은 Shift-화살표 키에) 적용되는 값이다. 이 값들은 윈도우 크기에 대한 % 값으로 지정된다. 예를 들어, 첫번째 값을 80% 로 지정하고 수직 스크롤바에서 마우스를 클릭하면 윈도우는 전체 화면의 80% 만큼 아래로 이동하게 된다. 즉, 20% 만큼 겹치는 부분이 생기는 것이다.

### 확대경

DVIWIN에는 돋보기 기능이 있다. 돋보기는 마우스의 가운데, 혹은 양쪽 버튼을 동시에 누르면 마우스 커서의 위치에 나타나며, 아무 버튼이나 다시 누르면 없어진다. 이 다이얼로그를 사용하여 돋보기의 크기를 지정할 수 있다. 이 중에서 **dynamic update**을 선택하면 돋보기에 보이는 화면이 움직임에 따라 계속 변하며, 그렇지 않으면 움직인 다음 왼쪽 마우스 버튼을 다시 눌러야 화면이 새로 변한다. 이 기능이 있는 이유는, **dynamic update**을 선택하지 않으므로써 본래의 화면과 확대된 화면을 비교할 수 없기 때문이다.

### 페이지 크기

이 다이얼로그를 사용하여 종이 크기를 선택할 수 있다. 처음에는 **8.5 x 11 inches**가 기본 크기로 지정되어 있다.

### 색깔 조정

이 명령어를 사용하여 확대 루틴에서 사용하는 색상 보간 (**interpolation**) 방식을 미세하게 조정할 수 있다. 이 값을 조정함에 따라 현재화면이 계속 바뀌므로 알맞는 값을 쉽게 찾을 수 있다.

### 폰트찾기

지정된 폰트가 없을 시에 어떤 조치를 취할지의 선택사항을 보여주는 다이아로그 박스를 나타내준다.

SoftTeX International

Mail address:

SoftTeX

3600 Wilshire Blvd. Suite 918,  
Los Angeles, CA 90010

## 5.7 참고 사항

dviwin.ini에서 정의되는 많은 변수들은 프로그램에서 대화 형식으로 변경될수있다. 특별히 중요한 변수값들은 다음과 같다.

0 fontpath : pk폰트가 있는 디렉토리를 지정한다. 다음 보기에서  
는 c:

yootex

pkdir

300pk의 디렉토리에서 300dpi 사이즈의 pk폰트를 찾아 dviwin이  
이용한다.

fontpath=C:\YOOTEX\pkdir\%rpk

0 pk-rst : 본 계수가 1일 때는 트루 타이프폰트 찾기에 앞서 pk폰  
트 디렉토리에서 먼저 해당하는 폰트를 찾는다.

0 hfont : 한글 폰트의 크기와 올림계수를 지정한다. 다음 보기에  
서는 한글폰트를 기본 크기에서 9% 크게하고 영어문자의 기준선  
에서 2% 내린다.

hfont=0.090000 0.020000

## 6장

# 원레이텍 포맷 설명

본장은  $\text{\LaTeX}$  version 2.09 에서 한글을 쓰기위해 확장한 원레이텍의 포맷에 대한 설명이다. 설명하지 않은 부분은  $\text{\LaTeX}$  version 2.09 의 사양을 따른다 (다음 장  $\text{\LaTeX}$  설명 참조).

### 6.1 $\text{\TeX}$ 에서의 한글 윈도우의 한글 폰트이용

현재의  $\text{\TeX}$  Engine은 1 바이트 문자 코드만을 다룰수 있으며 2 바이트 한글 코드를 다루기 위해서는 별도의 방법이 강구 되어야 한다.

두가지 방법이 가능한데 그 하나는 한글의 2 바이트 폰트를 여러개의 1바이트 폰트로 구분하여 코드컨버터로  $\text{\TeX}$  으로 컴파일 하기전에 전처리 하는것이다. 이때는 기존의 출력기 (dvi드라이버)를 그대로 이용할수 있는 장점이 있으나  $\text{\TeX}$ 에서 쓰는 폰트의 갯수가 지나치게 많아짐으로  $\text{\TeX}$ 의 용량을 제한한다.

다른 하나는 2byte 폰트이름을 그대로 이용한다. 이를 위해 반사이즈의 폰트미트릭화일 (tfm 화일)을 만들어 **TEX** 컴파일러가 이용하도록 한다. 그 2byte 정보가 **TEX**에 의하여 분리되지 않도록 전처리기를 통하여 매크로로 묶어주고 dvi 출력기에서 이 코드를 조합하여 해당하는 **typeface**를 만들어 낸다. 이 방법의 장점은 한글 윈도우에서 표시될수 있는 여러가지 문자, 기호를 그대로 출력할수 있다. (예: 일본어, 한자어, 기호) 윈레이텍은 이 방법을 택하였다. 윈레이텍에서 제공하는 한글 폰트 이름은 다음과 같다.

TeX tfm 화일 이름	한글 Window의 폰트 이름
hgs	중서체
hdr	굴림체
hgt	돋움체
hmj	바탕체
hmjb	굵은 바탕체
hmji	기운 바탕체
husrone	(사용자정의)

보기로서 plain.tex에 한글 폰트가 추가된 hwtex포맷에서의 한글 폰트 정의 화일의 내용은 다음과 같다.

```

\font\mj-v=hmj scaled 500
\font\mjsix=hmj scaled 600
\font\mjsev=hmj scaled 700
\font\mjegt=hmj scaled 800
\font\mjnin=hmj scaled 900
\font\mjten=hmj scaled 1000
\font\mjelv=hmj scaled 1100
\font\mjtwl=hmj scaled 1200
\font\mjfrtn=hmj scaled 1400
\font\mjsvtn=hmj scaled 1700

\font\mjtwty=hmj scaled 2000
\font\mjtwfv=hmj scaled 2500
%
\font\mjb-v=hmjb scaled 500
\font\mjbsix=hmjb scaled 600
\font\mjbsev=hmjb scaled 700
\font\mjbegt=hmjb scaled 800
\font\mjbnin=hmjb scaled 900
\font\mjbten=hmjb scaled 1000
\font\mjbelv=hmjb scaled 1100

```

```

\font\mjbtwl=hmjb scaled 1200
\font\mjbftrn=hmjb scaled 1400
\font\mjbsvtn=hmjb scaled 1700
\font\mjbtwty=hmjb scaled 2000
\font\mjbtwfv=hmjb scaled 2500
%
%\font\mji-v=hmji scaled 500
%\font\mjisix=hmji scaled 600
%\font\mjisev=hmji scaled 700
%\font\mjiegt=hmji scaled 800
%\font\mjinin=hmji scaled 900
%\font\mjiten=hmji scaled 1000
%\font\mjielv=hmji scaled 1100
%\font\mjitwl=hmji scaled 1200
%\font\mjiftrn=hmji scaled 1400
%\font\mjisvtn=hmji scaled 1700
%\font\mjitwty=hmji scaled 2000
%\font\mjitwfv=hmji scaled 2500
%
\font\gt-v=hgt scaled 500
\font\gtsix=hgt scaled 600
\font\gtsev=hgt scaled 700
\font\gtegt=hgt scaled 800
\font\gtnin=hgt scaled 900
\font\gtten=hgt scaled 1000
\font\gtelv=hgt scaled 1100
\font\gttwl=hgt scaled 1200
\font\gtftrn=hgt scaled 1400
\font\gtsvtn=hgt scaled 1700

\font\gttwty=hgt scaled 2000
\font\gttwfv=hgt scaled 2500
%
\font\gs-v=hgs scaled 500
\font\gssix=hgs scaled 600
\font\gssev=hgs scaled 700
\font\gsegt=hgs scaled 800
\font\gsnin=hgs scaled 900
\font\gsten=hgs scaled 1000
\font\gselv=hgs scaled 1100
\font\gstwl=hgs scaled 1200
\font\gsftrn=hgs scaled 1400
\font\gssvtn=hgs scaled 1700
\font\gstwty=hgs scaled 2000
\font\gstwfv=hgs scaled 2500
%
\font\dr-v=hdr scaled 500
\font\drsix=hdr scaled 600
\font\drsev=hdr scaled 700
\font\dregt=hdr scaled 800
\font\drnin=hdr scaled 900
\font\drten=hdr scaled 1000
\font\drelv=hdr scaled 1100
\font\drtwl=hdr scaled 1200
\font\drftrn=hdr scaled 1400
\font\drsvtn=hdr scaled 1700
\font\drtwty=hdr scaled 2000
\font\drtwfv=hdr scaled 2500

```

이론적으로는 투루타이프로 한글 윈도우에 설치되는 어떠한 폰트도 **WnTeX**에서 쓸수 있다. 먼저 **tex**에서 쓰는 **tfm**화일을 만들어야 한다.

영문의 경우는 **ttf2tfm**의 프로그램을 이용하여 투루타이프의 폰트 화일에서 **tfm**화일을 만들어 내고 한글은 박스문자로서 폰트간의 사이 간격 크기가 거의 같기 때문에 기존의 **mj.tfm**을 이름을 바꾸어 쓸수 있

다.

본 윈레이텍시스템에 새로운 폰트를 쓰고자 할때는 `dviwin.ini` 의 `usrfont1`의 줄에 사용자가 추가하고자 하는 한글폰트 이름을 아래와 같이 기입한다.

```
usrfont1=(한글폰트 이름)
```

이에 따라 TeX의 매크로를 아래와 같이 정의 해주어야 한다.

```
\font\mynewfont=usrone at 12pt
```

그다음에 TeX의 source화일에서 `mynewfont` 로서 새로 추가된 폰트를 선택할수 있다.

## 6.2 윈레이텍의 LaTeX(lplain)

윈레이텍의 LaTeX 포맷은 `hwlatex.tex` 에서 정의 되며 LaTeX에서도 한글 폰트도 TeX의 `plains` 와 같은 방법으로 설치되어 있다. 기본적으로 1992년 버전의 `lfont.s.tex`에 위 TeX에서 언급된 한글 폰트가 포함되어있고 자세한 그폰트의 선택방법은 `latex` 장을 참조하시요.

### 6.2.1 한글 counter

윈레이텍 LaTeX에서는 다음과 같은 두개의 한글 counter function을 정의 하였다. (`han{}`, `Han{}`, `hnum{}`)

포맷화일의 소스화일인 `hanenum.tex`을 보면 어떻게 새로운 문자열을 사용할수 있는지 알수있다. 기본적으로 `KS5601` 코드 테이블에 나와 있는 모든 문자열 이용할수있다. 다음 보기를 보시요.



```

\begin{list}{\Han{\value{mycnt}}}
{\usecounter{mycnt}}
\item 명칭 : 본회는 명칭은동산학교 동창
회라 칭한다.
\item 장소 : 본회는 동산학교안에 둔다.
\item 목적 : 본회는 회원간의 친목을 도모
하고
동산 학교의 부흥과 발전에 기여한다.
\end{list}

```

- ㉠ 명칭 : 본회는 명칭은동산학  
교 동창회라 칭한다.
- ㉡ 장소 : 본회는 동산학교안에  
둔다.
- ㉢ 목적 : 본회는 회원간의 친목  
을 도모하고 동산 학교의 부  
흥과 발전에 기여한다.

물론 han{} 과 Han{}의 최대 값은 13이다. (한글의 자음 갯수)

```

\begin{list}{\han{\value{mycnt}}}
{\usecounter{mycnt}}
\item 명칭 : 본회는 명칭은동산학교 동창
회라 칭한다.
\item 장소 : 본회는 동산학교안에 둔다.
\item 목적 : 본회는 회원간의 친목을 도모
하고
동산 학교의 부흥과 발전에 기여한다.
\end{list}

```

- ㉠ 명칭 : 본회는 명칭은동산학  
교 동창회라 칭한다.
- ㉡ 장소 : 본회는 동산학교안에  
둔다.
- ㉢ 목적 : 본회는 회원간의 친목  
을 도모하고 동산 학교의 부  
흥과 발전에 기여한다.

물론 han{} 과 Han{}의 최대 값은 13이다. (한글의 자음 갯수)

```

\begin{list}{\hnum{\value{mycnt}}}
{\usecounter{mycnt}}
\item 명칭 : 본회는 명칭은동산학교 동창
회라 칭한다.
\item 장소 : 본회는 동산학교안에 둔다.
\item 목적 : 본회는 회원간의 친목을 도모
하고
동산 학교의 부흥과 발전에 기여한다.
\end{list}

```

- ① 명칭 : 본회는 명칭은동산학  
교 동창회라 칭한다.
- ② 장소 : 본회는 동산학교안에  
둔다.
- ③ 목적 : 본회는 회원간의 친목  
을 도모하고 동산 학교의 부  
흥과 발전에 기여한다.



## 제 II 부

# LaTeX 설명



## 7장

# QUICK START:

## 한글 $\text{\LaTeX}$ 화일의 예

본장에서는 한글  $\text{\LaTeX}$  화일의 예를 통해  $\text{\LaTeX}$  사용에 대한 초보적인 지식을 얻고자 한다.

보기 1은 간단한 한글  $\text{\LaTeX}$  화일의 예이다. 이것은 원래  $\text{\LaTeX}$ 에 포함되어 있는 예제 `SMALL.TEX` 화일을 한글로 번역하여 약간 수정한 것이다. 첨부된 매뉴얼을 참조하여 사용자는 이 화일을 입력해 보기 바란다. 줄 왼쪽에 표시된 숫자는 알아보기 쉽게 하기 위해 씌여졌을 뿐 화일엔 포함되지 않는다. 보기 2는 이 화일을 처리하고 난 뒤의 결과이다.

대부분의 문서는 거의 전부가 본문으로 이루어졌다. 단어가 모여서 문장이 되고 문장이 모여 단락이 되며 단락이 모여 본문을 만든다. 앞에서 제시된 예도 예외는 아니다. 본문은 자연스럽게 타이핑함으로써 별 문제없이 입력된다. 그러나 이 문서가 출력될 때에는  $\text{\LaTeX}$ 이 알아서 줄을 채워주고 단어 사이의 간격도 조절해주며 양측 여백도 알맞게

```

1: %SMALL. TEX -- Released 5 July 1985 -- 한글번역 1991년 12월 30일
2: %이 화일을 입력화일의 예로 이용하십시오.
3: %한 줄에서 %이호다음에 나오는 모든 것들은
4: %LaTeX에 의하여 무시됩니다.
5: %
6: %경고! 다음의 10개의 기호는 그 것의 출력을 위하여 사용하면
7: %안됩니다:      & $ # % _ { } ^ ~ \
8:
9: \documentstyle[11pt,a4]{article} %입력화일은 이런형태의 두줄로
10: \begin{document}                %반드시 시작해야하고
11:                                  %맨 끝에 \end명령을 써 주어야
12:                                  %한다.
13: \section{짧은 글}                %이 명령은 절의 제목을 지정합니다.
14:
15: 단어 사이는 한 번 또는 그 이상 띄워도 무방합니다. 단락사이에는
16:   한개 이상의 빈 줄이 들어가야 합니다. 출력된 결과는 두개 이상의
17:   띄어쓰기나 두개 이상의 빈 줄에 의하여 영향을 받지 않습니다.
18:
19:
20: 큰 따옴표는 다음과 같이 입력합니다: ``인용구절''.
21: 작은 따옴표는 다음과 같이 입력합니다: `인용구절'.
22:
23: 긴 데쉬는 데쉬를 --- 처럼 세번 입력합니다.
24:
25: 강조할 구절은 다음과 같이 합니다: {\em 강조된 구절}.
26: 볼드로 쓸 구절은 다음과 같이 합니다: {\bf 볼드로 쓰인 구절}.
27:
28: \subsection{한 두가지 경고}      %이처럼 절의 제목을 만듭니다.
29:
30: 문장 도중에 사용된 마침표 뒤에 너무 넓은 공백이 생기면 (예를 들어
31: 기다라는 뜻의 영어인 et c. 의 뒤에는 문장이 끝날 때 만큼의 띄어쓰기는
32: 필요가 없다.) 괄호 안에서 처럼 마침표 다음에 \verb| |를 하면 됩니다.
33:
34: \$나 \verb| |와 같은 특수 기호를 출력하기 위하여 이것들을 직접
35: 입력하면 안됩니다! 그 중 일곱개는 아래와 같이
36: \verb| |와 함께 입력하면 됩니다: \$ \& \# \% \_ \{ and \}.
37: 그 밖의 것들을 어떻게 출력하는지에 대해서는 매뉴얼을 참조 하십시오.
38:
39: \end{document}                    %입력 파일은 반드시 이처럼 끝난다.

```

그림 7.1: 간단한 LaTeX 파일의 예

---

## 7.1 짧은 글

단어 사이는 한번 또는 그 이상 띄워도 무방합니다. 단락사이에는 한개 이상의 빈 줄이 들어가야 합니다. 출력된 결과는 두개 이상의 띄어쓰기 나 두개 이상의 빈 줄에 의하여 영향을 받지 않습니다.

큰 따옴표는 다음과 같이 입력합니다: \인용구절". 작은 따옴표는 다음과 같이 입력합니다: `인용구절`.

긴 데쉬는 데쉬를 | 처럼 세번입력합니다.

강조할 구절은 다음과 같이 합니다: *강조된 구절* 볼드로 쓸 구절은 다음과 같이 합니다: **볼드로 씌인 구절**.

### 7.1.1 한 두가지 경고

문장 도중에 사용된 마침표 뒤에 너무 넓은 공백이 생기면 (예를 들어 기타라는 뜻의 영어인 **etc.** 의 뒤에는 문장이 끝날 때 만큼의 띄어쓰기는 필요가 없다.) 괄호 안에서 처럼 마침표 다음에 \를 하면 된다.

\$나\와 같은 특수 기호를 출력하기 위하여 이것들을 직접 입력하면 안 됩니다! 그 중 일곱개는 아래와 같이 \와 함께 입력하면 됩니다: \$ & # % \_ f and g. 그 밖의 것들을 어떻게 출력하는지에 대해서는 매뉴얼을 참조 하십시오.

---

그림 7.2: 간단한 파일이 처리되어 출력되어진 모습

맞추어 준다. 단어 사이의 공백은 몇 개 이든지 최종 출력에는 아무런 영향이 없다. 왜냐하면 여러개의 공백이  $\text{\LaTeX}$ 에 의해 하나의 공백으로 처리되기 때문이며 또한 편집기에서의 줄끝도 단어 사이의 하나의 공백 처럼 처리된다. 새로운 단락의 시작은 입력파일에서는 하나의 빈 줄로 표시되어야 하고 따라서 단락을 시작할 때가 아니면 공백인 줄은 없어야 한다.  $\text{\LaTeX}$ 에서는 자판에 있는 몇 개의 글자를 그 글자를 출력하기 위하여 그대로 사용해서는 안된다. 다음 열 개의 글자

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

가 바로 그것들이다. 보기 1을 보면  $\backslash$  로 시작하는  $\text{\LaTeX}$  명령들이 있는데 (줄 9, 10, 13 참고), 이것들은  $\text{\LaTeX}$  명령이고, 문서의 구조에 관한 것들이다. 이러한 명령들에 대하여 사용자가 명심해야 할 몇 가지 사실이 있는데 이들은:

- 0 모든  $\text{\LaTeX}$  명령들은  $\backslash$  로 시작하고, 한 개 이상의 글자를 포함한다.
- 0  $\text{\LaTeX}$  명령은 알파벳만을 사용해야 하고 대문자와 소문자를 구별해서 사용해야 한다. 따라서  $\backslash \text{BEG N}$  은  $\backslash \text{begi n}$  과 다르다.
- 0 어떤 명령들은 문장 속에서 사용되는데 이들은 글자 형태를 바꾸는데 주로 쓰여진다.  $\backslash \text{em}$  이라는 명령은 강조하기 위해서 사용되는 것으로 한글에서는 고딕체로 출력되고 영어에서는 이탤릭체로 출력된다 (줄 25 참고). 이러한 명령들과 이 명령에 영향을 받아야 하는 말은 항상  $\{$  와  $\}$  로 둘러 싸여져야 한다. 예를 들어  $\{\backslash \text{em}$  는 강조하는 상태로 들어가는 것이고,  $\}$  는 이 상태에서부터 빠져 나오는 명령이 된다.
- 0 다음과 같은 명령들도 있는데



`\command{ 뭐뭐 }`

이 경우 '뭐뭐'는 이 명령의 인수이다. `\section`이 이러한 명령의 예이다(줄 13 참고). 대개의 경우 '{ }' 로 둘러싸으로써 인수를 표시 하지만 때로는 '[' ]'가 사용되기도 하고 두 종류가 함께 사용되는 경우도 있다. 이런 식으로 쓰는 것이 귀찮겠지만 사용자는 이에 따라야 한다.

0 명령이 순전히 인수없이 알파벳으로만 이루어진 경우에 `LaTeX`이 명령의 끝을 알기 위해서는 공백이나 알파벳이 아닌 글자가 나와야 한다.

`LaTeX`을 사용하여 작성되는 문서에 반드시 포함되어야 할 명령들이 있다. 본문은 `\begin{document}` 라는 명령으로 시작해야 하고 `\end{document}` 라는 명령으로 끝나야 한다(줄 10, 39 참고). 끝나는 명령 `\end{document}` 다음에 오는 것들은 모두 무시된다. `\begin{document}` 이라는 명령 전에 오는 것들은 전문(*preamble*) 이라고 불리우는 것인데 어떤 종류의 문서 형식을 사용할 것인가를 지정하는 명령들만을 포함할 수 있다.

전문에 반드시 포함되어야 하는 명령은 `\documentstyle` 으로 (줄 9 참고), 이것은 전체적인 문서형식을 결정한다. 보기파일은 간단한 보고서 형식의 문서로서 `article`이라는 문서형식을 사용하고 있으며, A4용지와 11pt 크기의 글자로 출력하기 위하여 이 명령이 보완되어 있다. 그 밖에 많은 다른 문서형식들이 있는데 이것은 나중에 소개하기로 한다.

`LaTeX`은 작은 따옴표 또는 큰 따옴표와 같은 열고 닫는 인용부호들도 물론 출력할 수 있는데 닫는 따옴표인 '를 출력하기 위해서는 자판에서는 4 또는 . 와 같이 되어있는 것을 사용하면 되고 (줄 21 참고), 여는 따옴표인 `는 자판에서는 3와 같이 생긴 것을 사용하면 된다(줄 20 참고). 큰 따옴표는 이것들을 두번 타이핑하여 얻을 수 있다. 따라서 큰

따옴표 " 는 사용하지 않아도 된다.  $\text{\LaTeX}$ 은 세 종류의 길이를 가지는 데쉬를 출력할 수 있는데 자판의 데쉬를 한 번, 두 번 또는 세 번 연속으로 타이핑함에 따라 점점 길어진다' (줄 23 참고). '10{20' 에서의 데쉬는 두 번 타이핑한 것이며 한 번 타이핑한 데쉬는 하이픈을 표시하기 위하여 주로 사용한다. 때로는 앞에서 언급한 것과 같은  $\text{\LaTeX}$ 이 고유 의 뜻을 가지는 심볼들을 출력할 필요가 있는데 이들 중 일곱 개는 \ 를 그 심볼앞에 사용하면 얻어진다(줄 36 참고). 나머지 세 개의 심볼과 y, z, x, \$, .c, ], | 와같이 자판에 나타나지않는 것들은 다른 명령들을 이용하여 출력할 수 있다.

$\text{\LaTeX}$ 화일에 출력을 원하지는 않지만 입력자의 편의를 도모하기 위한 주석이 포함되어야 하는 경우도있다. 한 줄에서 %기호 다음에 나오는 모든 것이  $\text{\LaTeX}$ 에 의하여 무시되기 때문에 이를 이용하여 주석을 달 수 있다.

$\text{\LaTeX}$ 에는 다음과 같이 네 개의 문서형식이 있다:

`article` 이라는 문서형식은 간단한 문서 또는 기사 형태로 출력하기 위한 것이다. 따라서 장을 표시하기 위한 명령이 없으며 `\maketitle` 를 이용하여 문서의 제목을 쓰면 (9절 참조)제목은 별도의 표지를 만들지 않고 첫 페이지의 상단에 나타나게 된다.

`report` 라는 문서형식은 보고서 형태로 출력하기 위한 것으로서 `article`과 비슷하나 장을 만들 수 있고, 제목은 별도의 페이지에 표시된다.

`book` 라는 문서형식은 책을 출판하기 위한 것으로서 양면 인쇄에 알맞게 페이지들을 출력할 수 있다.

`letter` 라는 문서형식은 사적인 서양 편지작성을 위한 것으로서 주소, 날짜, 서명들을 위한 명령이 제공된다.

이러한 표준 문서형식은 형식사양들에 의하여 보완될 수 있는데 `\documentstyle` 명령뒤의 '['의 안에 쓰면된다. 단 한 가지의 문서형식만이 사용되어야하지만 형식사양은 한 개 이상 사용할 수 있으며, 두 개 이상 사용되었을 경우에는 ', '에 의해 나누어져 있어야 한다. 표준 형식사양들은 다음과 같다:

`11pt` 라는 사양은 11pt짜리 크기의 글자로 본문이 출력되고 이러한 크기에 대한 사양이 없으면 10pt짜리 글자크기로 출력된다. 11pt짜리 글자는 10점짜리 글자보다 대략 10%정도 크다.

`12pt` 라는 사양은 본문을 12점짜리 크기의 글자로 출력하며 12pt짜리 글자는 10pt짜리 글자보다 20%정도 크다.

`twoside` 라는 사양은 `article` 또는 `report` 의 문서형식에서 양면 인쇄를 위한 형태로 만들어준다. 이 사양은 `book` 문서형식에서는 기본이다.

`twocolumn` 이라는 사양은 모든 페이지에서 두 개의 세로 단 형태로 출력하기 위한 것이다.

`titlepage` 는 `\maketitle` 라는 명령이 문서형식 `article`에서 별도의 표지에 제목을 쓰게 해주며 이러한 별도 표지는 `report` 와 `book`에서는 항상 나타난다.

원래의 `LaTeX`에는 없었지만 우리나라와 같이 A4용지를 주로 사용하는 나라에서 많이 사용되는 사양으로 `a4`라는 것이 있는데, 이는 `LaTeX`이 개발된 미국에서 주로 사용되는 용지가 A4보다 조금 짧고 폭은 조금 넓기 때문에 이 사양없이 A4용지에 출력하면 보기에 좋지않기 때문에 필요하다.



# 8장

## 글자입력

### 8.1 글자모양

TEX에서 사용가능한 글자모양은 다음과 같다.

`\rm` Roman type 한글 로만. 기본

`\bf`: Boldface **한글볼드**

`\it`: Italic *한글 이태릭*

`\sc`: Small caps

`\sf`: Sans serif

`\sl`: Slanted

`\tt`: Typewrite 한글 tt

`\em` Emphasized *한글강조*

`\mj`: 한글 명조. default

`\gt`: 한글 고딕

`\gs`: 한글 궁서

`\dr`: 한글 디나루

## 8.2 글자크기

문서의 초기 글자크기를 정하는 방법은 선택에 크기를 써주면 된다.

예) `documentstyle[ ] article` 일때

10pt : `\documentstyle[ ] { article }`

11pt : `\documentstyle[ 11pt ] { article }`

12pt : `\documentstyle[ 12pt ] { article }`

글자크기 조정명령의 예는 다음과 같다.

<code>\tiny</code>	English	한글 (漢字)
<code>\scriptsize</code>	English	한글 (漢字)
<code>\footnotesize</code>	English	한글 (漢字)
<code>\small</code>	English	한글 (漢字)
<code>\normalsize</code>	English	한글 (漢字)
<code>\large</code>	English	한글 (漢字)
<code>\Large</code>	English	한글 (漢字)
<code>\LARGE</code>	English	한글 (漢字)
<code>\huge</code>	English	한글 (漢字)

## 8.3 섞어 쓰기

`{\large\gt 크기}` 를  
`{\large\gs 먼저}` 정해주고  
`{\small\mj 글자모양}` 을  
`{\normalsize\dr 나중}` 에 쓴다.

크기를 먼저 정해주고 글자모양  
을 나중에 쓴다.

모든 글자와 크기가 가능한 것은 아니고 표 10.1 에서 O나 N으로 표시된 것만 가능하다. N으로 표시된 것은 일반적인 text mode에서는 O로 표시된 것과 똑같이 사용할 수 있지만, 수식을 쓸 경우에는 사용직전에 `\load{\normal size}{\sc}`와 같이 명령을 써야 한다.

## 8.4 특수기호의 출력

LaTeX은 자판에 없는 많은 기호들의 출력을 가능하게 해준다. 예를 들어, 모든 영어 알파벳에 다음과 같이 엑센트를 붙일 수가 있다.

$\text{\`{a}}$	$\text{\~{o}}$	$\text{\v{b}}$	$\text{\c{o}}$	$\text{\' {o}}$
$\text{\={o}}$	$\text{\H{o}}$	$\text{\d{o}}$	$\text{\^{o}}$	$\text{\. {o}}$
$\text{\t{oo}}$	$\text{\b{o}}$			
$\text{\" {o}}$	$\text{\u{o}}$			

그 밖에 다른 특수 기호들도 아래와 같은 명령들로 출력된다:

y	<code>\dag</code>	x	<code>\S</code>	.c	<code>\copyright</code>
z	<code>\ddag</code>	{	<code>\P</code>	\$	<code>\pounds</code>
<	<code>\oe</code>	?	<code>\OE</code>	;	<code>\AE</code>
>	<code>\AE</code>	&	<code>\aa</code>	&	<code>\AA</code>
=	<code>\o</code>	@	<code>\O</code>	A	<code>\l</code>
A	<code>\E</code>	:	<code>\ss</code>	>	<code>'</code>
<	<code>!`</code>	:::	<code>\l dots</code>	LaTeX	<code>\LaTeX</code>

이외에도 문자의 번호를 8진수로 직접 다음과 같이 지정할 수도 있다.

예를 들면 `backslash \` 의 경우에 다음과 같이 쓸 수 있다.

0 한번 쓰고 마는 경우 : `\symbol{'134}`

0 계속 쓸 경우 : 새로운 명령어로 지정.

{ 명령어 지정 방법 : `\newcommand{\bs}{\char'134}`

{ 사용방법 : 필요한 위치에서 `\bs` 라고 하면 된다.

자세한 문자표를 본 매뉴얼 끝장의 부록 A 를 참조 하기 바란다.



도표 8.1: 사용가능한 글자모양과 크기

O : 항상 사용가능

N : 수식에서 지정 필요

X : 사용 불가능

	<code>\rm</code>	<code>\it</code>	<code>\bf</code>	<code>\sl</code>	<code>\sf</code>	<code>\sc</code>	<code>\tt</code>
5pt	O	N	N	X	X	X	X
6pt	O	N	N	X	X	X	X
7pt	O	O	N	N	N	N	N
8pt	O	O	N	N	N	N	N
9pt	O	O	O	N	N	N	O
10pt	O	O	O	O	O	N	O
11pt	O	O	O	O	O	N	O
12pt	O	O	O	O	O	N	O
14pt	O	N	O	N	N	N	N
17pt	O	N	O	N	N	N	N
20pt	O	N	N	N	N	N	N
25pt	O	X	N	X	X	X	X



## 9장

# 수식 쓰는 방법

TeX의 최대장점 중의 하나는 수식이 아름답게 표현된다는 것이다. TeX이 원래 AMS<sup>1</sup>에서 만든 것이기 때문에 각종 잡지나 책에 나오는 수학기호를 기본적으로 제공해준다.

### 9.1 간단한 수식 표현

수식 표현은 주로 아래위 첨자와 분수의 표현 근호(root), 생략 기호와 수학기호, log 등으로 이루어져있다. 이 수식 모드에서의 영문자들은 이탤릭체로 나타나고 일반 텍스트보다 간격이 좀 넓다. 또한 영문자 사이의 빈 공간은 무시된다. 한가지 알아 둘 것은 수식에서는 한글을 쓸 수 없다는 것이다. 한글을 쓸 때에는 `\mbox` 명령을 사용한 텍스트로 입력한다.

1. math environment : `$ equation-text $` 형태

문장 중간에 수식이 나타날 때 사용하는 방법이다. 
$$\sum_{n=0}^1 x_n$$
 과

---

<sup>1</sup>American Mathematical Society : 미국 수학회

같은 수식이 나타날 때는 수식의 앞뒤에 \$ 를 붙여주면 된다. 간단한 경우에는 \$ 수식 \$로도 되지만 나중에 예러가 나오기 쉬운 부분이므로 주의하기 바란다. 이와 같은 기능으로 \ ( equation-text \ ) 또는 \ begin{mat h} equation-text \ end{mat h} 라고 해도 된다.

2. displaymath environment : \[ equation-text \] 아래와 같이 수식을 나타낼 때 사용하는 방법이다. 별 옵션이 없으면 가운데에 수식이 나타난다.

$$\sum_{n=0}^{\infty} x_n$$

이는 앞뒤에 \$\$ 를 붙여주거나 \[ equation-text \] 또는 \ begin{di spl aymat h} equation-text \ end{di spl aymat h} 와 같이 할 수 있다.

3. equation environment : \ begin{equat i on} equation-text \ end{equat i on}

이는 위에서 설명한 displaymath와 같다. 하지만 여기에서는 수식의 오른쪽 끝에 수식번호를 붙여준다. 아래의 수식과 같이 나오게 된다.

$$\sum_{n=0}^{\infty} x_n \tag{9.1}$$

이 번호는 각장(chapter)이나 article에서 자동으로 증가하게 된다.

참고로 fl eqn 옵션은 di spl aymat h} 나 {equat i on} environment에서 수식을 왼쪽으로 붙여준다. 이를 위해서는 \ document st yl e 명령을 다음과 같이 써주면 된다.

```
\ document st yl e[ 12pt , fl eqn] { art i cl e}
```

아래의 예는 자주 쓰이는 간단한 수식들의 이용 실례이다.

0 상하첨자

output	input
$x_2$	<code>\$x_2\$</code>
$2y_3$	<code>\$2y_3\$</code>
$x_2y_3$	<code>\$x_2y_3\$</code>
$x = y_{4z}$	<code>\$x=y_{4z}\$</code>
$x^2$	<code>\$x^2\$</code>
$x^2y^3$	<code>\$x^2y^3\$</code>
$x = y^{4z}$	<code>\$x=y^{4z}\$</code>
$\Gamma^{(1)}(R)$	<code>\$\Gamma^{(1)}(R)\$</code>
$x^{y^z}$	<code>\$x^{y^z}\$</code>

0 / 나 \frac에 의한 분수의 표현

$3=4$	<code>\$3/4\$</code>
$y = \frac{1}{x}$	<code>\$y=\frac{1}{x}\$</code>

\frac{ }{ }은 displaymath나 equation 또는 eqnarray에서 제대로 나타난다.

$y = \frac{x^2 + z = 2}{x^3 + x + 1}$	<code>\[y=\frac{x^2+z=2}{x^3+x+1}\]</code>
$V(r) = \frac{e^{-r}}{r} - \frac{1}{r^2}$	<code>\[V(r)=\frac{e^{-r}}{r}-\frac{1}{r^2}\]</code>
$P(E_i) = e^{-\frac{E_i}{kT}}$	<code>\[P(E_i)=e^{-\frac{E_i}{kT}}\]</code>

0 근호의 표시(\sqrt)

$$\begin{aligned}
 y &= \sqrt{x} & \$y=\sqrt{x}$ \\
 y &= \sqrt[n]{n+1} & \$y=\sqrt[n]{n+1}$ \\
 a_{i+1} &= \sqrt{a_i^2 + p a_i} & \$a_{i+1}=\sqrt{a_i^2+\sqrt{a_i}}$
 \end{aligned}$$

0 생략기호 : 이중에서 `\lvert \dot s`는 수식 밖에서도 사용할 수 있다.

종류	output	input
아래	$x_1; x_2; \dots; x_n$	<code>\$x_1, x_2, \lvert \dot s, x_n\$</code>
가운데	$1 + 2 + \dots + 99$	<code>\$1+2+\lvert \dot s+99\$</code>
수직	$\vdots$	<code>\$\dot s\$</code>
대각선	$\ddots$	<code>\$\ddot s\$</code>

## 9.2 복잡한 수식 표현

복잡한 수식을 표현할 때 보통 '='이 정렬된 형태로 나타나는 경우가 많다. `TEX`에서는 이를 자동으로 정렬해서 보기 좋게 표현하는 `environment`를 가지고 있다. 앞에서처럼 `\begin{eqnarray} \text{equation-text} \end{eqnarray}`를 써 주면 된다. 아래의 예는 전형적인 실례이다.

$$\begin{aligned}
 E(X^{i+n}; X > t) &: \frac{\int_t^{\infty} f(x; \lambda) dx}{(i-1)! \dots (i+n-1)!} \\
 &= \frac{\int_t^{\infty} \lambda^i x^{i+n-1} e^{-\lambda x} dx}{(i-1)! \dots (i+n-1)!} \\
 &= \frac{\lambda^n}{(i-1)! \dots (i+n-1)!} \int_t^{\infty} x^{i+n-1} e^{-\lambda x} dx
 \end{aligned}$$

```

\begin{eqnarray*}
E(X^{-n}, X > t) \approx \frac{\beta a^n}{(\alpha - 1) \dots (\alpha - n)} \int_t^\infty f(x) dx \\
\hat{\mu}, \hat{\sigma} \\
= \frac{\beta a^n}{(\alpha - 1) \dots (\alpha - n)} \\
\cdot \left[ \Phi \left( \frac{t - \hat{\mu}}{\hat{\sigma}} \right) \right. \\
\left. - \frac{\mu^n}{\prod_{i=1}^n (\mu^2 - i \sigma^2)} \cdot \right. \\
\left. \left[ \Phi \left( \frac{\mu t - \mu^2 + n \sigma^2}{\sigma \sqrt{\mu^2 - n \sigma^2}} \right) \right] \right]
\end{eqnarray*}

```

때로는 우변이나 좌변이 너무 길어서 이를 나누어 주어야 하는 경우가 있다. 이 경우에는 다음과 같이 한다.

```

\begin{eqnarray*}
x & = & a - b - c - d \\
& & \mbox{-} e - f - g
\end{eqnarray*}

```

$$x = a - b - c - d - e - f - g$$

여기에서 `\mbox{-}` 명령은 'e' 앞에서 '-' 부호를 쓰기 위해서 빈 box를 하나 만들어주기 위한 것이다. 이를 만들어 주지 않으면 '-'와 'e'가 너무 붙어버리게 된다. 문장 중에서 minus가 아닌 'negative e'를 만들기 위해서는 `{-e}`로 써주어야 한다. `\begin{eqnarray*}`에 '\*'를 붙인 것은 수식번호를 붙이지 말라는 뜻이다. '\*'를 붙이지 않으면 앞에서 나왔던 equation mode와 마찬가지로 자동으로 증가하는 번호를 오른쪽에 붙여준다. 한편 긴 수식을 두줄로 잘라주는 또다른 방법이 있다.

```

\begin{eqnarray*}
\lefteqn{x + y + z + a + b =} \\
& c + d + e + f + g
\end{eqnarray*}

```

$$x + y + z + a + b = c + d + e + f + g$$

앞서의 예와 같이 두번째 줄의 & 두개는 `c + d + e + f + g`를 오른쪽으로 보내주는 역할을 한다.

### 9.3 배열(array)

배열을 만들기 위해서는 `array environment`를 이용해야 한다. 각 행(column)은 (l,c,r)에 의해서 왼쪽, 가운데, 오른쪽에 위치하게 된다. 각 열(row)은 `\\`에 의해서 분리된다. 열(row)안에서 행(column)을 분리하기 위해서는 `&`를 각 항 사이에 넣어주어야 한다. 이 `array`는 수학기호에서만 사용할 수 있다. 이는 뒤에 `tabular environment`와 유사한데 10.2를 참조하기 바란다. 아래의 예를 보자.

```

\[\begin{array}{ccc}
a & 14 & c \\
d-3 & e & f \\
g & h & \lambda \\
\end{array}\]

```

$$\begin{array}{ccc}
 a & 14 & c \\
 d-3 & e & f \\
 g & h & \lambda
 \end{array}$$

여기에서 `ccc`는 세개의 행이 모두 가운데 위치하게 됨을 의미한다. row 사이에 `\\`이 들어있고 각 행 사이에 `&`이 들어 있음을 알 수 있다. 아래는 좀더 복잡한 배열의 예이다.

```

\[\det \left|
\begin{array}{cccc}
x_0 & x_1 & x_2 & \cdots & x_n \\
x_1 & x_2 & x_3 & \cdots & \\
& x_{n+1} & \\
x_2 & x_3 & x_4 & \cdots & \\
& x_{n+2} & \\
\vdots & \vdots & & \ddots & \\
\vdots & \ddots & \vdots & & \\
x_n & x_{n+1} & x_{n+2} & \cdots & x_{2n} \\
& \cdots & x_{2n} \\
\end{array}
\right| > 0
\]

```

$$\det \begin{array}{cccc}
 - & - & - & - \\
 - & X_0 & X_1 & X_2 & \cdots & X_n & - \\
 - & X_1 & X_2 & X_3 & \cdots & X_{n+1} & - \\
 \det & - & X_2 & X_3 & X_4 & \cdots & X_{n+2} & - > 0 \\
 - & \vdots & \vdots & \vdots & \ddots & \vdots & - \\
 - & \vdots & \ddots & \vdots & & & - \\
 - & X_n & X_{n+1} & X_{n+2} & \cdots & X_{2n} & - \\
 & & & & \cdots & x_{2n} & 
 \end{array}$$

위의 예에서는 배열밖에 또다른 항이 있으며 `det`를 나타내기 위하여 `j`를 배열 양쪽에 붙여 놓았다. 위의 두개의 input 예에서 보듯이 모두 `displaymath` 모드안에 `\begin{array} \end{array}`가 들어있다. 물론 이 `array environment`를 `equation environment`에서도 실행가능하다.



여기에서 또한 짚고 넘어가야 할 것이 `delimiter`의 설정이다. `delimiter`란 양쪽에 붙는 괄호를 통칭하는 것이다. 이 `delimiter`를 쓸 때는 반드시 `\left`와 `\right`를 양쪽에 붙여주어야 한다. 양쪽에 붙는 것은 서로 다를 수 있으며 부록 A.8에 나오는 모든 기호를 쓸 수 있다. 한쪽만 쓰고자 하는 경우는 반대편에 `\left`. 또는 `\right`.와 같이 점(period)을 찍어 주면 된다. 이렇게만 해주면 `delimiter`는 자동으로 높이를 조정해 준다. 아래의 예를 보고 참고하기 바란다.

```

\left[ \left[ \begin{array}{c}
\left( \begin{array}{cc}
a & b \\
c & d-1 \\
e & f
\end{array} \right) \\
y \\
2z
\end{array} \right] \right]

```

또 하나의 유용한 명령은 `multicolumn`, 문자 정렬, 테두리 긋기이다. 아래의 예를 보자.

```

\begin{array}{| r@{.}l@{.} |}
\hline
12 & 345 & 321 & 645 \\
3985 & 34 & 85763 & 346 \\
\multicolumn{2}{| c{x} & 234 & 356 \\
987 & 82745 & & \\
\multicolumn{2}{| c{ }{\mbox{unk.}} \\
234 & 425 & \multicolumn{2}{| c{ }{y} \\
\hline
\end{array}

```

12:345	321:645
3985:34	85763:346
x	234:356
987:82745	unk.
234:425	y

이 `array`는 소수점을 중심으로 정렬된 배열을 보여준다. `TEX`에서는 이를 제공하지 않기 때문에 이런 차선책을 쓰는 것이다. 위의 예에서 `@.}`으로 표시된 점을 중심으로 숫자들이 정렬된다. 물론 이자리에는 다른 글자가 들어 갈 수도 있다. `\begin{array}` 뒤에 붙어나오는 `|`는 세

로줄을 긋는 것을 말한다. 마찬가지로 `\hrule` 은 가로줄을 긋는다. 즉 `\` 뒤에 `\hrule`을 쓰게되면 그 아래에 가로줄을 긋는다. 어떤 내용이 하나 이상의 `column`에 걸쳐있을 때 `\multicolumn{...}`을 이용한다. 첫 번째 `f g`에 들어가는 것이 몇개의 `column`에 걸쳐있는가를 나타낸다. 두 번째 `f g`에 있는것이 그 내용이 놓이는 위치이고(`r,c,l`), 세번째가 들어갈 내용이다. `\nbox{}` 명령은 `f g` 안의 내용이 일반적인 `text mode(roman type)`로 처리된다는 것을 의미한다.

## 9.4 기호및 기타

TEX에서 쓸 수 있는 수학기호들은 수식모드에서만 쓸 수 있다. 예를 들면,  $\alpha$  를 쓰기 위해서는 수식모드임을 나타내는 `$` 를 앞뒤에 붙여 `$\alpha$`라고 써야한다. 또한 반드시 `\` 다음에 나타나야 한다. 예를 들어  $y^2 \neq 4$ 를 나타내기 위해서는 `$y^{2}\not = 4$`라고 써야 한다. 한편 TEX는 네개의 간격 조정 명령을 제공한다. `\,` (조금), `\:` (중간정도), `\;` (많이), `\!` (약간좁힌다)이다.

예)

```
input $A\ , B\ : C\ ; D! E$
```

```
output A B C DE
```

책 뒤의 부록에 `\log`류 "함수 list"가 나와 있다. 이들은 수식중에서 자동으로 `roman`으로 나타난다. 수식중에서 다른 문자들을 `roman`으로 나타내기 위해서는 간단히 `\rm`을 써주면 된다. 예를 들면 `{\rm sinc}`와 같이 쓰면 된다.

한편 이 기호들을 쌓는 방법이 있다. `stackrel{...}{...}`이 그것인데, 예를 들자면  $\overset{!}{+}$ 와 같은 symbol은 `$\stackrel{\!}{\right arrow}{+}$`이라고 입력함으로써 가능하다. 수식의 위아래에 줄을 긋기 위해서는 `\overline{ }`

이나 `\underline{ }` 을 쓰면된다. 이는 수식뿐 아니라 일반 `text mode` 에서도 가능하다. 다음과 같은 `\widehat{ }` 과 `\widetilde{ }` 과 `\underbrace{ }` 과 `\overbrace{ }` 기능도 있다.

$$\begin{array}{l}
 \$\widehat{A+B} \quad = \quad \widehat{A+B} \\
 \widetilde{AB} \$ \quad \backslash \\
 \$\overbrace{y^2} + \underbrace{y+z+3} \$ \\
 \text{brace}\{y+z\} + 3 \$
 \end{array}$$

`\choose`도 알아두면 유용한 명령이다. 이는 다음과 같이 쓴다.

$$\$_m C_n = \{m \choose n\} \$ \quad {}_m C_n = \frac{m!}{n!}$$

TEX에서는 `caligraphic`문자를 수식에서 제공하는데 `alphabet` 대문자 26자가 있다. 이를 쓰려면 `\cal F`와 같이 하면 된다.

$$\begin{array}{l}
 \backslash [\backslash cal F] = \\
 - \frac{\partial \backslash cal V}{\partial r} \backslash \\
 F = ! \frac{\textcircled{A}}{\textcircled{B}}
 \end{array}$$

## 9.5 수식의 사용의 실례

이들을 위한 `input`은 그림과 같다.

## 9.6 수식출력의 예

이번 보기에서는 `theorem`, `corollary`, `lemma` 등의 사용 예를 보여준다.

수식을 출력할 수 있는 환경에는 세 가지가 있는데 `\(나 $로 시작하여 \)`나 `$로 끝나는 수식환경`과 `\[나 $$로 시작하여 \]`나 `$$로 끝나는 수식 전시환경`과 이 밖에 등식 환경이 있다. 수식환경은 문장속에서 수식을 쓸 때 사용되며 수식 전시환경은 주로 긴 수식이나 강조하고 싶은 수식을 새로운 줄의 중앙에 전시할 필요가 있을 때 사용된다. 등식환경

그림 9.1: 수식의 사용의 실례

1.

$$w^{\$} = \lim_{n \rightarrow \infty} \frac{f^n(\#_o)! \#_o}{n}$$

2.

$$P + kx + r V(x) = A \cos(\omega t)$$

3.

$$Q = 2e^{i\omega(x)}; Q^{\$} = 2e^{i\omega(x)}$$

4.

$$R_n = \frac{x^n}{n!} \# m(m-1)\dots(m-n+1)$$

5.

$$\int_{z_1}^{z_2} (z - z_0)^n dz = a_n \frac{(z - z_0)^{n+1}}{n+1} \Big|_{z_1}^{z_2}$$

6.

$$\sinh z = \sum_{n=1, \text{ odd}} \frac{z^n}{n!} = \sum_{s=0} \frac{z^{2s+1}}{(2s+1)!}$$

7.

$$\frac{z^4}{a+b+z} = \frac{z^4}{z+y+z} = A z^0 + B z^1 + C z^2$$

8.

$$U(x, y, z) = \begin{matrix} 3 & 4 & 2 & 4 \\ e^{i(x+y+z)} & e^{i(x-y+z)} & \cos(x-y+z) & \sin(x-y+z) \\ 0 & e^{i(x+y-z)} & \sin(x-y+z) & \cos(x-y+z) \\ 3 & & & 4 \\ e^{i(x+y-z)} \cos(x-y+z) & e^{i(x-y+z)} \sin(x-y+z) & & \end{matrix}$$

9.

T(23)		E	3C <sub>2</sub>	4C <sub>3</sub>	4C <sub>3</sub> <sup>0</sup>
(R <sub>x</sub> ; R <sub>y</sub> ; R <sub>x</sub> ) <sup>0</sup> (x; y; z)	A	1	1	1	1
	E	1	1	!	!
	T	3	!	0	0

$$! = e^{2: i=3}$$

## 그림 9.2: 입력화일

```

1. \[ w^* = \lim_{n \rightarrow \infty} \frac{f^n(\theta_0) - \theta_0}{n} \]
2. \[ \ddot{x} + k \dot{x} + \nabla V(x) = A \cos(\omega t) \]
3. \[ Q = \eta e^{i \phi(x)}, Q^* = \eta e^{-i \phi(x)} \]
4. \[ R_n \leq \frac{x^n}{n!} \times m(m-1) \cdots (m-n+1) \]
5. \[ a_n \int (z-z_0)^n dz = \int \frac{a_n}{(z-z_0)^{n+1}} dz \]
6. \[ \sinh z = \sum_{n=1, \text{odd}} \frac{z^n}{n!} = \sum_{s=0} \frac{z^{2s+1}}{(2s+1)!} \]
7. \[ \underbrace{a + \overbrace{b + \cdots + y}^{24} + z}^{26} \rightsquigarrow A \stackrel{a}{\rightarrow} B \stackrel{b}{\rightarrow} C \]
8. \begin{eqnarray*}
\{ \alpha, \beta, \gamma \} \\
& = & \left( \begin{array}{c} e^{i \gamma / 2} \\ 0 \\ e^{i \gamma / 2} \end{array} \right) \\
& \left( \begin{array}{c} \cos \beta / 2 \ \sin \beta / 2 \\ -\sin \beta / 2 \ \cos \beta / 2 \end{array} \right) \\
& \left( \begin{array}{c} e^{i \alpha / 2} \\ 0 \\ e^{i \alpha / 2} \end{array} \right) \\
& = & \left( \begin{array}{c} e^{i(\alpha + \gamma) / 2} \cos \beta / 2 \\ e^{i(\alpha - \gamma) / 2} \sin \beta / 2 \\ -e^{-i(\alpha - \gamma) / 2} \sin \beta / 2 \\ e^{-i(\alpha + \gamma) / 2} \cos \beta / 2 \end{array} \right)
\end{eqnarray*}
9. \begin{eqnarray*}
\begin{array}{c|c|ccc} \hline
\multicolumn{2}{c}{T(23)} & E & 3C_2 & 4C_3 & 4C'_3 \\ \hline
& A & 1 & 1 & 1 & 1 \\
& E & \left( \begin{array}{c} 1 \\ 1 \end{array} \right) & \left( \begin{array}{c} 1 \\ 1 \end{array} \right) & \left( \begin{array}{c} 1 \\ 1 \end{array} \right) & \left( \begin{array}{c} 1 \\ 1 \end{array} \right) \\
& \left( \begin{array}{c} \omega \\ \omega^2 \end{array} \right) & \omega & \omega^2 & \omega & \omega^2 \\
\left( \begin{array}{l} R_x, R_y, R_x \end{array} \right) & (x, y, z) & & & & \\
\left( \begin{array}{l} T & 3 & -1 & 0 & 0 \end{array} \right) & & & & & \\
\left( \begin{array}{l} \omega = e^{2\pi i / 3} \end{array} \right) & & & & & \\
\end{eqnarray*}

```

```

1: % 수식이 포함된 한글LaTeX 입력 파일의 예 { 한국과학기술원 수학연구센터
2:
3: \documentstyle[11pt]{article}
4: \newtheorem{theorem}{정리} %[section]
5: \newtheorem{corollary}{theorem}[따름정리]
6: \newtheorem{lemma}[theorem]{보조정리}
7: \newenvironment{pf}{\medskip\noindent{\bf
명.}\enspace}{\h-\ll\newline\smallskip}
8: \setcounter{section}{2}
9:
10: \begin{document}
11: \section{월리스의 곱}
12: 우리의 처음 목표는 월리스의 곱으로 알려진 다음 극한을 구하는 것이다.
13: \begin{theorem} $$\frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n}{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n+1)}$$
14: \frac{45}{65} \frac{67}{2n} \cdot \frac{2n}{2n+1} \frac{2n}{2n+1}$$
15: \end{theorem}
16: \begin{pf} 이 증명은 다음과 같이 연습문제로 남겨둔다.
17: \begin{enumerate}
18: \item sine함수의 거듭제곱의 적분에 관한 점화식을 사용하여 다음을 증명하라.
19: $$\int_0^{\pi/2} \sin^{2n} x \, dx = \frac{2n-1}{2n} \int_0^{\pi/2} \sin^{2n-2} x \, dx$$
20: $$\int_0^{\pi/2} \sin^{2n+1} x \, dx = \frac{2n}{2n+1} \int_0^{\pi/2} \sin^{2n-1} x \, dx$$
21: \item $n = 1, 2, 3, \dots$ 에 대하여 $\sin^n x$가 감소한다는 사실과 위의 정적분들을
22: 이용하여 다음 부등식이 성립함을 보여라.
23: $$\int_0^{\pi/2} \sin^{2n} x \, dx \leq \frac{1}{2n} \int_0^{\pi/2} \sin^{2n-2} x \, dx \leq \frac{1}{2n} \int_0^{\pi/2} \sin^{2n-4} x \, dx \leq \dots \leq \frac{1}{2n} \int_0^{\pi/2} \sin^2 x \, dx \leq \frac{1}{2n} \int_0^{\pi/2} 1 \, dx = \frac{\pi}{4n}$$
24: \item 구간 $[0, \pi/2]$에서의 $\sin^{2n} x$와 $\sin^{2n+1} x$의 정적분의 비를 취하여
25: 월리스의 곱을 유도하라.
26: \end{enumerate}
27: \end{pf}
28: \vspace{25pt}
29: \begin{corollary} $$\lim_{n \rightarrow \infty} \frac{(n!)^{2/2n}}{(2n)!^{1/2n}} = \pi^{1/2}$$
30: \end{corollary}
31: \begin{pf} 월리스의 곱은 $\displaystyle \frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n}{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n+1)}$와 같이 바꾸어 쓸 수 있다.
32: $\frac{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n}{1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n+1)} = \frac{2 \cdot 4 \cdot 6 \cdot \dots \cdot 2n}{2 \cdot 3 \cdot 4 \cdot \dots \cdot (2n+1)}$와 같이 바꾸어 쓸 수 있다.
33: 이 식의 양변의 제곱근을 취하면 원하는 식이 얻어진다.
34: \end{pf}
35: 마지막으로 다음 식이 성립함을 보이고 스텔링의 식에 들어있는 상수 $c$의 값이
36: $\frac{1}{\sqrt{2\pi}}$임을 보여라.
37: \begin{eqnarray*}
38: c &= & \lim_{n \rightarrow \infty} \frac{(2n)^{2n} e^{-2n}}{(2n)!} \\
39: &= & \lim_{n \rightarrow \infty} \frac{(n!)^{2/2n}}{(2n)!^{1/2n}} \\
40: \left[ \frac{n^{n+1/2} e^{-n}}{n!} \right]^2 &= & \sqrt{2\pi} c^2.
41: \end{eqnarray*}
42: \end{document}

```

증

그림 9.3: 수식이 포함된 LaTeX 화일의 예

## 9.7 윌리스의 곱

우리의 처음 목표는 윌리스의 곱으로 알려진 다음 극한을 구하는 것이다.

정리 1

$$\frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2 \cdot 2 \cdot 4 \cdot 4 \cdot 6 \cdot 6 \cdots 2n \cdot 2n}{1 \cdot 3 \cdot 3 \cdot 5 \cdot 5 \cdot 7 \cdots (2n-1)(2n+1)}$$

증명. 이 증명은 다음과 같이 연습문제로 남겨둔다.

1. sine함수의 거듭제곱의 적분에 관한 점화식을 사용하여 다음을 증명하라.

$$\int_0^{\pi/2} \sin^{2n} x dx = \frac{2n-1}{2n} \int_0^{\pi/2} \sin^{2n-2} x dx; \quad \int_0^{\pi/2} \sin^{2n+1} x dx = \frac{2n}{2n+1} \int_0^{\pi/2} \sin^{2n-1} x dx$$

2.  $n = 1; 2; 3; \dots$ 에 대하여  $\sin^n x$ 가 감소한다는 사실과 위의 정적분들을 이용하여 다음 부등식이 성립함을 보여라.

$$\int_0^{\pi/2} \sin^{2n-1} x dx < \int_0^{\pi/2} \sin^{2n+1} x dx < \frac{1}{2n}$$

3. 구간  $[0; \pi/2]$ 에서의  $\sin^{2n} x$ 와  $\sin^{2n+1} x$ 의 정적분의 비를 취하여 윌리스의 곱을 유도하라.

따름정리 2

$$\lim_{n \rightarrow \infty} \frac{(n!)^2 2^{2n}}{(2n)! n^{1/2}} = \pi$$

증명. 윌리스의 곱은  $\frac{\pi}{2} = \lim_{n \rightarrow \infty} \frac{2 \cdot 2 \cdot 4 \cdot 4 \cdots (2n-2)(2n-2)}{1 \cdot 3 \cdot 3 \cdot 5 \cdots (2n-1)(2n+1)}$ 와 같이 바꾸어 쓸 수 있다. 이 식

의 양변의 제곱근을 취하면 원하는 식이 얻어진다.

마지막으로 다음 식이 성립함을 보이고 스텔링의 식에 들어있는 상수  $c$ 의 값이  $c = \frac{\pi}{2}$ 임을 보여라.

$$\begin{aligned} c &= \lim_{n \rightarrow \infty} \frac{(2n)^{2n+1/2} e^{-2n}}{(2n)!} \\ &= \lim_{n \rightarrow \infty} \frac{(n!)^2 2^{2n} \frac{\pi}{2}}{(2n)! n^{1/2}} = \frac{\pi}{2} c^2 \end{aligned}$$

그림 9.4: 수식이 포함된 파일이 처리되어 출력되어진 모습

은 여러 줄의 등식의줄을 맞추어 쓸 필요가 있을 때 주로 사용되며 식의 오른쪽에 자동적으로 참조 번호를 붙여준다. 이러한 수식환경 안에서는 모든 알파벳이 이탤릭체로 되는 것이 기본이며 띄어쓰기가 무시되므로 보통의 문장을 이러한 환경안에서 쓰기 위해서는 `\nbox`라는 명령의 인수로써 그 문장을 입력해야한다.

### 9.7.1 기초적인 수식

첨수를 표시하기 위한 올려쓰기와 내려쓰기는 각각 `^`와 `_`를 사용하여보기 3의 줄 19에서 처럼 쓰면 되는데 여기서 `\int`는 적분기호를 나타내고 아래쓰기 `0`과 올려쓰기 `=2`가 사용되고 있다. 또한 줄 32에서는  $2^2$  등의 입력 방법이 나타나있다.

분수는 줄 29에서와 같이 `/`를 사용하여  $1=2$ 처럼 출력하거나 줄 13에서처럼 `\frac`명령을 사용하면 되는데 `\frac`는 다음에 나오는 두개의 인수를 취하여 처음 것은 분자 다음 것은 분모로 처리한다.

근호는 줄 40에서처럼 `\sqrt`를 사용한다. 다만  $n$ 승근을 표시하기 위하여는 `\sqrt[n]` 명령을 대신 사용하면된다.

생략을 표시하기 위한 점들은 줄 14와 같이 `\cdot s`를 사용하면 줄 중간에 점 세개가 찍히고 줄 21에서처럼 `\l dot s`를 사용하면 줄 아래 쪽으로 점 세개가 찍힌다. 이 밖에도 세로로 점이 찍히는 `\v dot s`와 대각선 방향으로 점이 찍히는 `\d dot s`가 있다.

### 9.7.2 수학기호

수학기호에는 많은 것들이 있기 때문에 여기서 모두 다루지는 않고 한 가지씩의 예를 들어봄으로써 대신하고자 한다. 여러분은 이를 출력하기 위하여 원 매뉴얼을 참고하기 바란다.



그리스문자는 문자의 영문이름을 `\`로 시작하여 쓰면 되는데 줄 13에서 `\pi`는 `:`를 출력한다.

많은 연산자 기호를 위한 한가지 예를 들면 `'`은 `\pm`으로 얻는다.

관계자 기호 전후에는 자동으로 약간의 공백이 들어간다. 관계자 기호를 위한 한 예는 줄 23의 `\le`이다. 이 `\le`는 5를 얻는다.

많은 화살표를 위한 명령이 있는데 그 중 한 예는 `\Rrightarrow`이다. 이 명령은)를 얻는다.

함수기호를 위한 명령은 줄 13의 `\lim`과 같은 것이 있는데 이것은 수식 안이지만 명조체로 `lim`를 출력한다. 이 함수기호에 수식 전시환경에서 올려쓰거나 내려쓰기를하면 보기 4에서 보듯이 그 기호의 위나 아래 (기호옆 위나 아래가 아닌) 첨자가 붙는다.

등식환경은 `\begin{eqnarray}` 명령과 `\end{eqnarray}` 명령을 사용하여 얻을 수있다. 이 때 줄을 맞추고자 하는 곳에 `&`를 써주면된다. 이 예는 줄 37에서 줄 41을 참고하기 바란다.



# 10장

## Table과 Tabbing

`tabbing` : `\tab` 을 친 것처럼 데이터들을 정렬시켜주는 방법

`tabular` : 말 그대로 가로세로 줄을 그어놓은 표를 작성.

### 10.1 tabbing 환경

`tabbing` 은 무엇보다 아래의 예를 보는 것이 빠르다. 아래와 같이 정렬할 때 `tabbing`을 이용한다.

보기입력은:

```
\begin{tabbing}
\small
Stock No. \= Description \= Wholesale Price \= Retail Price\
24 \> Disc Driver \> \$55.00 \> \$92.00 \
32 \> Monitor \> \$89.00 \> \$132.00 \
```

```
48 \> Keyboard \> \$77.00 \> $99.00
\end{tabbing}
```

그 결과는:

```
Stock No. Description Wholesale Price Retail Price
24      Disc Driver $55.00          $92.00
32      Monitor      $89.00          $132.00
48      Keyboard     $77.00          $99.00
```

"TAB stop" 이 첫줄에서 \=로 정해진다. 가장 왼쪽의 column은 자동으로 왼쪽끝으로 가서 출력된다. 각 column은 \> 으로 구분된다. 즉, 처음에 \=로 TAB을 쳤을때 어디에서 멈출것인가를 정해주고, 다음부터 \=로 TAB을 한번씩 치면서 내용을 써준다고 생각하면 된다. 주의할 것은 마지막 줄을 제외한 각줄의 끝에 \\을 넣어야 한다는 것이다.

이와는 달리 column의 폭을 사용자의 취향에 맞게 조절할 수 있는 방법이 두가지 있다.

첫번째는 아래와 같이 문자 x를 채우고 \kiii을 써주는 방법.

```
Stock No.      Description      Wholesale Price      Retail Price
24             Disc Driver      $55.00               $92.00
32             Monitor         $89.00               $132.00
48             Keyboard        $77.00               $99.00
```

이때 문자를 너무 적게 넣으면 아래와 같은 사태가 생긴다.

```
Stock No. Description Wholesale Retail Price
24      Disc Drive$55.00  $92.00
32      Monitor $89.00  $132.00
48      Keyboard$77.00  $99.00
```

두번째는 아래와 같이 각 `tab stop`의 위치를 직접 수치로 정해주는 것이다.

```
\begin{tabbing}
\hspace{2.5cm}\ = \hspace{4cm}\ = \hspace{4cm}\ = \kern 11
{\it Stock No.} \> {\it Description} \>
{\it Wholesale Price} \> {\it Retail Price} \\
24 \> Disc Driver \> \$55.00 \> \$92.00 \\
32 \> Monitor \> \$89.00 \> \$132.00 \\
48 \> Keyboard \> \$77.00 \> \$99.00
\end{tabbing}
```

tabbing에서 쓸 수 있는 명령과 기능.

0 \> : 다음 `tab stop`으로.

0 \< : 이전 `tab stop`으로.

0 \` : 오른쪽 끝으로. 단어의 오른쪽 끝을 맞추어준다.

0 \' : 이 표시 앞에 나온 단어를 이전 `column`의 왼쪽끝에 붙인다.

0 \ = : 새로운 `tab stop setting`.

0 \ \ : 새로운 행을 시작.

0 \kern 11 : 현재 `setting`을 바꾸지않고 출력하지 않은 채 새로운 행을 시작.

0 \+ : 새로운 행에서 첫번째 단어가 시작하는 위치를 `tab` 하나만큼 이동.

0 \- : 위의 \+ 와 반대로 동작.

0 \pusht abs : tabbing environment에서 현재의 tab 설정치 저장.

0 \popt ab : tabbing environment에서 \pusht ab으로 저장된 setting 복구.

## 10.2 tabular 환경 { 표 그리기

tabular environment에서 사용되는 명령이나 option.

0 tabular format option : | 는 세로줄, c는 center, l은 left, r은 right 를 나타낸다.

0 \hline : 가로줄 긋기.

0 \cline{m:n} : 가로줄 긋기. n번째 column에서 m번째 column까지.

0 multicolumn{l}{format option} : 현재 column에서 l개의 column을 하나의 column으로 취급. 현재행에서만 국부적으로 작용.

0 : & 는 column을 구분. \\ 는 새로운 줄로.

보기입력은:

```
\begin{tabular}{|l|l|r|} \hline
gnat s      & gram      & \$13.65 \\ \cline{2-3}
              & each      & .01 \\ \hline
```

```

gnu      & stuffed & 92.50 \\
\cline{1-1} \cline{3-3}
emur    &      & 33.33 \\ \hline
armadillo & frozen & 8.99 \\ \hline
\end{tabular}

```

그 결과는:

gnats	gram	\$13.65
	each	.01
gnu	stu, ed	92.50
emur		33.33
armadillo	frozen	8.99

보기입력은:

```

\begin{tabular}{|l|l|l|r|} \hline \hline
{\em type} & \multicolumn{2}{c}{} \\
{\em style} \\ \hline
smart      & red & short \\
rather silly & puce & tall \\ \hline \hline
\end{tabular}

```

그 결과는:

type	style	
smart	red	short
rather silly	puce	tall

## 10.3 Table과 Figure 환경

사용방법

```
\begin{figure}[position] 또는 \begin{table}[position]
```

내용

```
\caption[lot entry]{figure or table caption}
```

```
\end{figure} 또는 \end{table}
```

독립적인 table이나 -gure를 만들고 번호가 들어가는 caption(표제)를 붙이는 데 이용된다. two-column에서 양쪽 column에 다 걸치는 table이나 -gure를 만들기 위해서는 `\begin{figure*}` 또는 `\begin{table*}` 로 해주어야 한다. 다음은 위에서 사용된 옵션들 [ ]에 들어 있는 내용이다.

0 position : table이나 -gure가 오는 위치

h : here. 지금 위치에 바로 그림. table\*나 -gure\*에서는 쓸 수 없다.

t : top. page의 맨 위에

b : bottom. page의 맨 아래에. table\*나 -gure\*에서는 쓸 수 없다.

p : page of . oat. 따로 한 page를 차지한다.

예를 들면 `\begin{table}[tbp]` 와 같이 쓸 수 있다. default는 [tbp]이다.

0 \caption{ } : 내용 전후에 붙일 수 있는데 보통 table은 내용전에 붙이고 -gure는 내용다음에 붙인다.



0 lot entry : 이는 list of table(표 목차 또는 그림목차)을 만들 경우에 여기에 나타나는 내용. default는 { }안의 내용이다.

0 \caption{figure or table caption} : 그림이나 표의 표제. 절대 길게 쓰지 않도록 한다. 표 10.1의 input은 다음과 같다.

보기입력은:

```
\begin{table}[hbt]
```

```
\caption{사용가능한 글자모양과 크기} \label{font}
```

O : 항상 사용가능

N : 수식에서 지정 필요

X : 사용 불가능 \

```
\begin{tabular}{|c|c|c|c|c|c|c|c|} \hline
```

```
&\verb|\rm| &\verb|\it| &\verb|\bf| &\verb|\sl| &
```

```
\verb|\sf| &\verb|\sc| &\verb|\tt|\ \ \ \hline
```

```
5pt & O & N & N & X & X & X & X \ \ \hline
```

```
6pt & O & N & N & X & X & X & X \ \ \hline
```

```
7pt & O & O & N & N & N & N & N \ \ \hline
```

```
8pt & O & O & N & N & N & N & N \ \ \hline
```

```
9pt & O & O & O & N & N & N & O \ \ \hline
```

```
10pt & O & O & O & O & O & N & O \ \ \hline
```

```
11pt & O & O & O & O & O & N & O \ \ \hline
```

```
12pt & O & O & O & O & O & N & O \ \ \hline
```

```
14pt & O & N & O & N & N & N & N \ \ \hline
```

```
17pt & O & N & O & N & N & N & N \ \ \hline
```

```
20pt & O & N & N & N & N & N & N \ \ \hline
```

```

25pt & O & X & N & X & X & X & X \\ \hline
\end{t abul ar}
\end{t abl e}

```

그 결과는:

도표 10.1: 사용가능한 글자모양과 크기

O : 항상 사용가능 N : 수식에서 지정 필요 X : 사용 불가능

	<code>\rm</code>	<code>\it</code>	<code>\bf</code>	<code>\sl</code>	<code>\sf</code>	<code>\sc</code>	<code>\tt</code>
5pt	O	N	N	X	X	X	X
6pt	O	N	N	X	X	X	X
7pt	O	O	N	N	N	N	N
8pt	O	O	N	N	N	N	N
9pt	O	O	O	N	N	N	O
10pt	O	O	O	O	O	N	O
11pt	O	O	O	O	O	N	O
12pt	O	O	O	O	O	N	O
14pt	O	N	O	N	N	N	N
17pt	O	N	O	N	N	N	N
20pt	O	N	N	N	N	N	N
25pt	O	X	N	X	X	X	X

# 11장

## 여러가지 문단 정렬 방법

사용방법 : `\begin{ name} . . . \end{ name}` 으로 글의 앞뒤에 써주면 된다. `name`의 위치에 들어갈 수 있는 정렬방법(Formatting Environment)은 다음과 같다.

`center` : 가운데로 정렬

`. ushleft` : 왼쪽끝 맞춤

`. ushright` : 오른쪽끝 맞춤

`itemize` : 항목별로 정렬

`enumerate` : 번호를 붙여서 항목별로 나열

`description` : 단어를 설명하는 모양

`quote` : 인용 { 한 문단

`quotation` : 인용 { 두 문단 이상

verse : 시를 쓸때

verbatim : 화면에 나타나는 대로 출력. { 영어에만 해당.

## 11.1 center, . ushleft, . ushright

기본적인 input은 다음과 같다. \\은 줄바꿈

```
\begin{name}  
Reason, \\ when used for discovery \\ rather than defense, \\  
is an intellectually honest enterprise \\  
and hence one of life's greatest pleasures.  
\end{name}
```

위에 나타난 것과 같은 input - le에서 name을 center, . ushleft, . ushright로 써준 결과는 각각 다음과 같다.

### 1. center

```
Reason,  
when used for discovery  
rather than defense,  
is an intellectually honest enterprise  
and hence one of life's greatest pleasures.
```

### 2. . ushleft

```
Reason,  
when used for discovery
```

rather than defense,  
 is an intellectually honest enterprise  
 and hence one of life's greatest pleasures.

3. . ushright

Reason,  
 when used for discovery  
 rather than defense,  
 is an intellectually honest enterprise  
 and hence one of life's greatest pleasures.

## 11.2 itemize, enumerate, description

다음은 itemize의 예이다.

<code>\begin{itemize}</code>	0 center
<code>\item center</code>	
<code>\item . ushleft</code>	0 . ushleft
<code>\item . ushright</code>	0 . ushright
<code>\item[1.] itemize</code>	1. itemize
<code>\item[2.] enumerate</code>	2. enumerate
<code>\item[3.] description</code>	3. description
<code>\item[] quote</code>	quote
<code>\item[] quotation</code>	quotation
<code>\item 기타 { verse</code>	0 기타 { verse
<code>\end{itemize}</code>	

이 item들은 내부에 또다른 item을 가질 수 있다. 이는 item뿐 아니라 enumerate에서도 마찬가지이다. 다음은 enumerate의 예이다.

```

\begin{enumerate}
\item First sentence in the enumerate example
\begin{enumerate}
\item See the number noticeably
\item Can you see it?
\begin{enumerate}
\item See the number noticeably
\item Can you see it?
\end{enumerate}
\end{enumerate}
\end{enumerate}
\item Example of enumerate environment
\end{enumerate}

```

1. First sentence in the enumerate example

(a) See the number noticeably

(b) Can you see it?

i. See the number noticeably

ii. Can you see it?

2. Example of enumerate environment

다음은 `description`의 예이다.

```

\begin{description}
\small
\item[자귀나무] 함수초과의 낙엽 활엽 소교목. 산기슭의 양지에 남. 잎이 밤에는 오그라들며, 여름에는 분홍꽃이 피고 가을에 협과가 익음. 나무는 도구나 세공재임.
\item[자귀풀] 콩과의 1년초. 밭이나 습지에 나며 높이 80cm. 줄기는 연하고 속이 빈 원주형. 여름에 노란 꽃이 핌. 잎은 차 대용.
\end{description}

```

자귀나무 함수초과의 낙엽 활엽 소교목. 산기슭의 양지에 남. 잎이 밤에는 오그라들며, 여름에는 분홍꽃이 피고 가을에 협과가 익음. 나무는 도구나 세공재임.

자귀풀 콩과의 1년초. 밭이나 습지에 나며 높이 80cm. 줄기는 연하고 속이 빈 원주형. 여름에 노란 꽃이 핌. 잎은 차 대용.

### 11.2.1 Custom Description Lists

사용자가 `description list format`을 정의 해서 쓰고 싶을 때는 `preamble`에 아래와 같은 새로운 `format`을 정의해 주어야 한다.(`preamble`이란

`\documentstyle`과 `\begin{document}` 사이의 부분을 뜻한다. 12.4 참조.)

```
% namelist generates a list with an item width of
% your choice; form \begin{namelist}{width}
\newcommand{\namelistlabel}[1]{\nbox{#1}\hrfill}
\newenvironment{namelist}[1]{%
\begin{list}}
{
\let\makelabel\namelistlabel
\settowidth{\labelwidth}{#1}
\setlength{\leftmargin}{1.1\labelwidth}
}
}{%
\end{list}}
```

사용자가 정해주는 environment는 `\newcommand`와 `\newenvironment` 명령을 사용해서 만들 수 있다. 이는 17.2과 17.2에서 상세히 설명될 것이다. 이를 이용한 `input-le`은

<pre> \begin{namelist}{xxxxxxx} \item[{\gt 자귀나무}] 함수초과의 낙엽 활 엽 소교목. 산기슭의 양지에 남. 잎이 밤에는 오그라들며, 여름에는 분홍꽃이 피고 가을에 협과가 익음. 나무는 도구나 세공재임. \item[{\gt 자귀풀}] 콩과의 1년초. 밭이나 습지에 나며 높이 80cm. 줄기는 연하고 속이 빈 원주형. 여름에 노란 꽃이 핌. \end{namelist} </pre>	<p>자귀나무 함수초과의 낙엽 활엽 소교목. 산기슭의 양지에 남. 잎이 밤에는 오그라들며, 여름에는 분홍꽃이 피고 가을에 협과가 익음. 나무는 도구나 세공재임.</p> <p>자귀풀 콩과의 1년초. 밭이나 습지에 나며 높이 80cm. 줄기는 연하고 속이 빈 원주형. 여름에 노란 꽃이 핌. 앞은 차 대응.</p>
---	---

주의할 것은 `\begin{namelist}`에서 뒤에 붙은 `f g`에 넣은 `x`로 `item`의 폭을 조정했다는 것이다. 이 때의 출력은

### 11.3 quote, quotation

인용문을 쓸때 이용하는 방법. 왼쪽, 오른쪽끝이 조금씩 들어가서 출력되고 크거나 글자꼴이 `\begin{quote}`와 `\end{quote}`안에 국한되어 작용한다. `quotation`을 쓰는 방법은 `quote`와 같으나 두문단 이상을 인용할 때 이용한다. 문단을 나누는 요령은 앞장에서 설명한 바 있다. 앞 페이지의 예문은 다음과 같이 작성한 것이다.

영화 속에서 소리와 풍경의 어울림은 절묘하다. `만다라'에서 깨달음을 찾는 승려들의 대사가 길이나 풍경에 깔리는 배경음악처럼 일반 사람들에게 낯선 것이었던 데 비해 여기서 소리와 풍경은 서로 어우러져 그것만으로도 우리나라를 느낄 수 있을 만큼 감동적이다.



그러나 소리와 사람의 어우러짐에는 문제가 있다. 유봉과 송화의 관계는 한의 대물림이다. 그것은 표면적으로 소리의 대물림으로 나타나게 마련이다.

이를 위한 input은 다음과 같다.

\begin{quote}

영화 속에서 소리와 풍경의 어울림은 절묘하다.

‘만다라’에서 깨달음을 찾는

승려들의 대시가 길이나 풍경에 깔리는 배경음악처럼

일반 사람들에게 낯선 것이었던 데 비해

여기서 소리와 풍경은 서로 어우러져 그것만으로도

우리나라를 느낄 수 있을 만큼 감동적이다.

그러나 소리와 사람의 어우러짐에는 문제가 있다.

유봉과 송화의 관계는 한의 대물림이다.

그것은 표면적으로 소리의 대물림으로 나타나게 마련이다.

\end{quote}

## 11.4 verse { 시를 쓸때

시는 다른 문장과는 달리 문단이 바뀌어도 indentation이 생기지 않고 문단과 문단 사이를 약간 띄우는 것이 관례이다.

남으로 창을 내겠소

김 상 용

남으로 창을 내겠소.

밭이 한참같이

팽이로 파고

호미론 풀을 매지요.

구름이 꼬인다 갈 리 있소.  
새 노래는 공으로 들으랴오.  
강냉이가 익걸랑  
함께 와 자셔도 좋소.  
왜 사냐건  
웃지요.

이를 위한 input은 다음과 같다.

```
\begin{center}
{\large 남으로 창을 내겠소}
\end{center}
\begin{flushright}
{\gs 김~ 상 용~~~~~}
\end{flushright}
\begin{verse}
남으로 창을 내겠소.\\
밭이 한참갈이\\
괭이로 파고\\
호미론 풀을 매지요.

구름이 꼬인다 갈 리 있소.\\
새 노래는 공으로 들으랴오.\\
강냉이가 익걸랑\\
함께 와 자셔도 좋소.

왜 사냐건\\
웃지요.
\end{verse}
```

## 11.5 verbatim과 verb

이런 설명서나 program을 인용하고자 할 때는 가능한 한 원래의 꼴을 그대로 나타내주어야 한다. 이때는 TeX의 특수문자도 포함되는 경우가 많다. 이를 화면에 나타나는 그대로 출력하게 하는 것이 `verb`나 `verbatim`이다.

기본적인 사용 방법은 `\verb` 내용과 같이 써주면 된다는 것이다. 여기에 서 | 대신 다른 것을 써줄 수도 있는데 내용안에 그 문자가 포함되어서는 안된다는 것이 중요하다. 아래와 같이 특수문자가 포함될 때

```
# $ % & * {} ___ \ This all thing# $ % & * {} ___ \ This all thing
# $ % & * {} ___ \ This all thing# $ % & * {} ___ \ This all thing
```

이를 나타내기 위해서는:

```
\verb: # $ % & * {} ___ \ This all thing:
\verb+# $ % & * {} ___ \ This all thing+
\verb: # $ % & * {} ___ \ This all thing:
\verb># $ % & * {} ___ \ This all thing>
```

등과 같이 써주면 된다.

위에서 설명한 `verb`는 한 줄 이상은 쓸 수 없다. 한 줄 이상 써주기 위해서는 `verbatim environment`를 써야한다. 아래의 출력을 보라.

```
#include<stdio.h>
main() /* count lines in input */
{
    int c, nl;
    nl = 0;
    while((c = get char()) != EOF)
        if (c == '\n')
            ++nl;
    printf("%d\n", nl);
}
```

```
}
```

이를 얻기 위해서는

```
\begin{verbatim}
#include<stdio.h>
main() /* count lines in input */
{
    int c, nl;

    nl = 0;
    while((c = get char()) != EOF)
        if (c == '\n')
            ++nl;
    printf("%d\n", nl);
}
\end{verbatim}
```

와 같이 입력하면 된다.

# 12장

## 큰 글 짜기

이 장에서는 장(`chapter`), 절(`section`)을 나타내는 방법과 문서 구성에 대해 설명한다.

### 12.1 section 명령어

```
\chapter{chapter title}
\section{section title}
\subsection{subsection title}
\subsubsection{subsubsection title}
```

예)

이 page 의 `chapter`와 `section`은 다음과 같이 한 것이다.

```
\chapter{큰 글 짜기}
이 장에서는 장(chapter), 절(section)을 나타내는 방법과 문서 구성에 대해 설명한다. \section{section 명령어}
```

주의 : `\chapter` 명령은 `documentstyle`이 `article`일때는 쓸 수 없다. 위의 12.1과 같은 번호를 붙이지 않으려면 다음과 같이한다.

```
\subsection*{subsection title}
```

어느 경우든 `{ }`안에 있는 내용이 목차에 나타난다. 목차에 다른 내용이 나타나도록 하려면 다음과 같이한다.

```
\subsection*[contents title]{subsection title}
```

이때 `[ ]`안의 `contents title`은 목차에, `{ }`안의 `subsection title`는 내용중에 나타난다.

## 12.2 Title and Titlepage

`title`은 `\begin{document}` 바로 다음에 아래와 같이 써주면 된다.

```
\title{TeX Quick Reference}
```

```
\author{Man A and Man B}
```

```
\date{\today}
```

```
\maketitle
```

위의 `title`의 항목들은 자동으로 가운데 놓인다. `title`에는 3개의 정보가 있다. 만일 저자나 날짜를 생략하고 싶으면 `{ }`를 비워두면 된다. 비록 비워두는 한이 있어도 3개의 항목은 다 써야한다. `\maketitle` 명령어가 `title`을 작성해준다. `book`과 `report style`에서는 `title`을 별도의 `page`에 작성해준다. `article style`에서는 다음과 같이 써야 별도의 `page`에 만들어 줄 것이다.

```
\documentstyle[11pt, titlepage]{article}
```

`article style`에서는 `\maketitle` 앞에 `abstract`를 넣을 수 있다. 이는 아래와 같이 작성한다.

```
\begin{abstract}
```

The abstract's text is typed here.

```
\end{abstract}
```

## 12.3 목차 { Table of Contents

목차는 `table page`와 1장 사이에 아래와 같이 써준 곳에 나타난다.

```
\newpage
\pagenumbering{num_style}
\tableofcontents
```

`\newpage` : 목차를 새 page에 만든다. '다음장으로'라는 명령  
`\pagenumbering{num_style}` : page번호를 `num_style`에 따라 붙인다. 여기에  
들어갈 수 있는 것은

`arabic` : arabia 숫자.  
`roman` : 소문자 로마숫자.  
`Roman` : 대문자 로마숫자.  
`alph` : 소문자 alphabet.  
`Alph` : 대문자 Alphabet.

목차의 내용은 `.aux` 파일에 있는 `data`에 근거한 것이므로 목차가 나타나게  
하려면 `\tableofcontents`가 있는 `-le`에 대해서 `TEX`를 두번이상 실행시켜야  
한다.

그림목차나 표목차를 넣기 위해서는 `\tableofcontents` 뒤에 다음 명령을  
넣어주면 된다.

```
\listoffigures
\listoftables
```

### 12.3.1 머릿말

머릿말은 한 장(`chapter`)로 취급되어야 하면서 번호는 붙지않고 이것이 목차에 들어가야 한다. 때문에 다음과 같이 `\tableofcontents` 뒤에 써주어야 한다.

```
\chapter*{Preface}
\addcontentsline{toc}{chapter}{Preface}
\pagestyle{myheadings}
\markboth{Preface}{Preface}
Preface starts from here.
```

그리고 나서 다음번 `\chapter` 명령어 다음에

```
\pagestyle{headings}
```

라고 해 주어야한다. `header` 명령은 12.4.3에서 자세히 다룬다.

## 12.4 Preamble

이는 `input -le`의 앞부분 즉, `\begin{document}` 명령어 앞에 놓이는 갖가지 명령어들을 일컫는다. 이책의 `preamble`은 다음과 같다. %뒤에는 `comment`가 온다.

```
\documentstyle[han, emi nes, makei dx]{book}
%emi nes for picture format
\makeindex
%namelist generates a list with an itemwidth of
%your choice; form \begin{namelist}{width}
\newcommand{\namelistlabel}[1]{\nbox{#1}\hflll}
\newenvironment{namelist}[1]{%
\begin{list}{}}
{
```



```

\let\nakelabel\namelistlabel
\settowidth{\labelwidth}{#1}
\setlength{\leftmargin}{1.1\labelwidth}
}
}{%
\end{list}}
\def\baselinestretch{1.2} % 1.2 spacing option
\textwidth=6in
\textheight=8.5in
\topmargin=0pt
\oddsidemargin=0pt
\evensidemargin=0pt

```

### 12.4.1 Document Style Options

`\begin{document}` 의 [ ]안에 놓이는 것이 option이다. 여기에 쓰일 수 있는 것은 다음과 같다.

- 0 11pt 또는 12pt : 문서전체의 글자크기 { default 10pt.}
- 0 twocolumn : 2 column으로 문서작성.
- 0 twoside : 양면에 출력하기 위한 option.
- 0 titlepage : article에서만 사용가능( 12.2 참조)
- 0 makeindex : index 작성.

### 12.4.2 단면 그리고 양면 출력

단면 출력 : 종이의 한면만 사용한 경우의 출력.

양면 출력 : 종이의 양면 사용. 이 경우에는 page가 나타나는 위치가 홀수면과 짝수면이 다르다. 이 책이 그 예이다.

- 0 article, report : default 단면출력.

0 book : default 양면출력

### 12.4.3 header

header는 page에서 본문 위쪽에 나타는 내용으로 page번호와 section이 나타나는 곳이다. 머릿말에 나타는 내용은 documentstyle에 따라 다르지만 아래표와 같다.

documentstyle	단면 출력	양면 출력
book 과 report	chapter, page	짝수면 : chapter, page 홀수면 : section, page
article	section, page	짝수면 : section, page 홀수면 : subsection, page

이상은 default이고 다른 option도 있다. 이를 이용하려면 \pagestyle{option}을 preamble에 써주면 된다. 아래 4개의 option중 하나를 넣을 수 있다.

0 plain : header 없고, page를 footer에 붙인다. report, article style의 default option

0 empty : header와 footer가 없다. page가 나타나지 않는다

0 headings : 위에서 설명한대로. book style의 default option

0 myheadings : 사용자가 정의

위의 사용자정의 즉, myheading을 이용하려면, preamble에서 다음과 같이 머릿말을 정하면 된다.

0 단면 출력의 경우

```
\pagestyle{myheadings}
```

```
\markright{쓰고자하는 내용}
```

0 양면 출력의 경우

```
\pagestyle{myheadings}
```

`\marginbot h{ 왼쪽에 쓰고자하는 내용}{ 오른쪽에 쓰고자하는 내용}`

header는 document 첫 장과 chapter가 시작하는 page에는 나타나지 않는다.

## 12.4.4 Page Layout Option

input -le의 preamble에 써줌으로써 실제 페이지에서 글자가 나타나는 모양과 위치를 정할 수 있다. 그림으로 고쳐야 할 부분을 나타내어 보면 `single column`은 그림 10.1, `double column`은 그림 10.2에 나타난 것과 같다. `\command=length`로 써주면 된다. 예를 들면

```
\documentstyle[12pt]{article}
\headheight=0mm
\headsep=5mm
\oddsidemargin=1mm
\evensidemargin=1mm
\textheight=226mm
\textrightwidth=175mm
\def\baselinestretch{1.2}
\begin{document}
```

과 같이 해준다. `\baselinestretch`는 정해주는 방법이 다른 것과 좀 다르다. 그 이유는 `\baselinestretch`는 길이가 아니고 어떤 숫자이기 때문이다. default는 1로 되어있다. 일반적으로 `double spacing`을 해주려면 `\def\baselinestretch{2}`로 쓰면 된다. 여기에서 주의 할 것은 `\baselinestretch 12pt`은 예외적으로 `\begin{document}`이 후에 써야한다는 것이다. 그 뒤에 나오는 줄부터 줄간격이 바뀐다.

다음의 보기를 보시오.

```
% 두 줄 띄기
\newcommand{\DS}{\renewcommand{\baselinestretch}{2}
\tiny\normalsize}

% 한 줄 띄기
```

```
\newcommand{\SS{\renewcommand{\baselinestretch{1}}
\tiny \normalsize}
```

문단을 처음 시작할 때 들여쓰기를 정해주는 방법은 그림에 나타나지 않는 데 다음과 같은 명령이 있다.

0 \parindent 10mm: 들여 쓰기를 10mm로 정해준다.

0 \noindent : 들여쓰기를 하지 않는다. 주의할 것은 이는 preamble에 써 주는 것이 아니고 들여쓰기를 하지 않고자 하는 문단 시작직전에 써 주어야 한다는 것이다. 그러면 이는 그때만 들여쓰기를 하지 않는다는 것을 의미한다. 만일 문장 전체에 대해서 들여쓰기를 하지 않으려면 간단히 \parindent 0mm로 해주는 것이 좋다.

## 12.5 Appendix { 부록

\appendix라고 쓰면 된다. 그 이후에 \chapter, \section등의 명령어를 쓰면 'Appendix A', 'Appendix B'와 같이 번호가 나타나고 목차에도 마찬가지로 마찬가지다.

## 12.6 큰 문서작성

큰 문서를 작성할 경우 한꺼번에 한 화일에 저장하는 것은 바람직하지 않다. 이 경우 \input{inputfilename} 또는 \include{inputfilename}를 이용하여 다음과 같은 main -le을 만드는 것이 바람직하다.

\include{inputfilename}는 \includeonly{inputfilenames}로서 포함되는 화일을 선택하므로 부분 컴파일을 하기 쉽다. 확장명을 쓰지 않으면 자동으로 .tex -le을 읽어 들인다. .htx -le의 경우에는 자동으로 .tex -le로 전환해서 읽어 들인다. 이 책의 main -le은 다음과 같다.

```
\documentstyle[softinc]{hbook}
%\documentstyle[softinc,fullpage]{hbook}
```

```

\makeindex
\input{manstyle}
%includeonly{general}
%
% Layout
%
% 185 235
\setlength{\textwidth}{11cm}
\setlength{\textheight}{16.5cm}
\setlength{\topmargin}{-1.0cm}
\setlength{\headheight}{1.0cm}
\setlength{\headsep}{0.0cm}
\setlength{\footheight}{1.5cm}
\setlength{\footskip}{2.5cm}
\setlength{\oddsidemargin}{0cm}
\setlength{\evensidemargin}{2.5cm}
\pagestyle{fancyplain}
\addtolength{\headwidth}{\marginparwidth}
\addtolength{\headwidth}{\marginparwidth}
%
% FancyHeading Styles
%
\renewcommand{\chaptermark}[1]{%remember chapter title}
\renewcommand{\sectionmark}[1]{%}
\renewcommand{\headrule}{}
\lhead[{\fancyplain}]{\fancyplain}
\rhead[{\fancyplain}]{\fancyplain}
\renewcommand{\chaptermark}[1]{%remember chapter title}
\renewcommand{\sectionmark}[1]{%}
\renewcommand{\headrule}{}
\lfoot[{\fancyplain}]{\sf \thepage} \ {\sf 1.0}]{\fancyplain}
\rfoot[{\fancyplain}]{\fancyplain}{\sf } \ {\sf \thepage}
\cfoot{}

```

```

\pagenumbering{roman}
\biibliographystyle{plain}

%
% My Macros
%
%newcommand{\XXX}{} %%Comments
\newcommand{\yooindex}[1]{\index{#1} #1}
\newcommand{\menubox}[1]{\fbox{\strut\tt \index{!#1} #1}\ }
\newcommand{\button}[1]{\vbox to 0.55cm{\special{isocal e #1.bmp, 0.8cm 0.8cm}}\hspace
\newcommand{\shortcut}[1]{\ovalbox{\tt Ctrl-#1}}
%????????????macro to put a box around the text %????????????????????
%newcommand{\figbox}[3]{\hbox to #2cm{\vbox to #3cm{\special{isocal e #1.wrf, #2cm #
%
\newcounter{mycnt}

\begin{document}
\renewcommand{\baselinestretch}{1.5}\small \normalsize

%
%Title & preambles
%

\biibliographystyle{plain}
\pagenumbering{roman}
\title{\Huge\gs }
\author{Soft\TeX}
\date{1995 11 25}
%\maketitle
\tableofcontents
%listoffigures
%listoftables
\newpage

```

```

\pagenumbering{arabic}

\part{}
  \include{general}
  \include{spec}
  \include{install}
% XXX{
  \include{tutorial}
  \include{wintex}
  \include{htmlatex}
  \include{tex2doc}
  \include{dviwin}
  \include{hmacr}
%   \include{books}
% bibliography{./manual}

% newcounter{mycnt}
\part{\LaTeX}
  \include{quick}
  \include{typeset}
  \include{math}
  \include{table}
  \include{formath}
  \include{org}
  \include{bibliography}
  \include{index}
  \include{twocol}
  \include{style}
  \include{special}
  \include{error}
\part{Samples}
\renewcommand{\baselinestretch}{0.8}
  \include{sample}

```

```

% \include{set up}
\appendix

\renewcommand{\baselinestretch}{1.5}
\include{msymbol}
\include{ref}
\ bibliography{./manual}
\printindex
% %XXX
\end{document}

```

\begin{document}와 \end{document}가 들어가는 것에 주의할 것. input-  
-le에는 \begin{document}, \end{document}가 들어가지 않아야

한다. 이와 같이 다른 -le을 불러오는 명령은 아래와 같은 것이 있다.

0 \input{input filename}: input-le을 이 위치에 삽입. page를 자동으로  
넘기지 않음.

0 \include{input filename}: \input{ }과 유사. 자동으로 page넘김.

0 \newpage: 수동으로 page를 넘김. 다만 \twocolumn에서는 column만 바  
꾼다.

0 \clearpage: \newpage와 유사. \twocolumn에서도 page바꿈

0 \clearpage: page를 넘겨서 홀수면(오른쪽면)에서 시작.



# 13장

## Bibliography의 작성

### 13.1 Bibliography Database 파일 : BIB 파일

0 BIB 파일 이름의 extension은 \.bib"

0 BIB 파일 포맷

저자, 제목, 출판사 등의 문헌 정보들을 각 문헌별로 묶어서 열거한 ASCII 파일

```
@type{key,  
  fi el d1 = " .... ",  
  fi el d2 = " .... ",  
  .  
  .  
  .  
  fi el dn = " .... ",  
}
```

0 예)

```
@BOOK{texbook,  
  author = "Donald E. Knuth",  
  title = "The {\TeX} book",  
  publisher = "Addison-Wesley",  
  year = 1984 }
```

```
@BOOK{lathex,  
  author = "Leslie Lamport",  
  title = "{\LaTeX\rm} {A} Document Preparation System",  
  publisher = "Addison-Wesley",  
  year = 1986 }
```

.  
.

- 0 **key**는  $\text{\LaTeX}$ 에서 특정 문헌을 인용할 때 **Label**로서 사용된다.
- 0 이 후의 **-eld**들은 순서없이 열거될 수 있다.
- 0 저자가 둘 이상 있는 경우에는 반드시 **\and**로 구분하여야 한다. 중 괄호는 특정 단어를 **BibTeX**이 소문자로 바꾸지 않고 대문자로 유지하게 하는데 사용된다.

## 13.2 Entry Types

**article** Journal, magazine의 article.

필수 피일드 : author, title, journal, year

옵션 피일드 : volume, number, pages, month, note

**book** 출판사가 있는 책

필수 피일드 : author 또는 editor, title, publisher, year

옵션 피일드 : volume 또는 number, series, address, edition, month, note

**booklet** 출판사가 없는 책

필수 피일드 : title

옵션 피일드 : author, howpublished, address, month, year, note

conference inproceedings와 동일

**inbook** 책의 일부분

필수 피일드 : author 또는 editor, title, chapter 또는(외) pages, publisher, year

옵션 피일드 : volume 또는 number, series, type, address, edition, month, note

**incollection** 책의 일부분 중 제목이 붙은것

필수 피일드 : author, title, booktitle, publisher, year

옵션 피일드 : volume 또는 number, series, type, chapter, pages, address, month, note

**inproceedings** conference proceeding의 한 article

필수 피일드 : author, title, booktitle, year

옵션 피일드 : editor, volume 또는 number, series, pages, organization, publisher, address(conference 개최 장소), month, note

**manual** 매뉴얼

필수 피일드 : title

옵션 피일드 : author, organization, address, edition, month, year, note

**mastersthesis** 석사 학위 논문

필수 피일드 : author, title, school, year

옵션 피일드 : type, address, month, note

**misc** 맞는 type이 없을 때

필수 피일드 :

옵션 피일드 : author, title, howpublished, month, year, note

**phdthesis** 박사 학위 논문

필수 피일드 : author, title, school, year

옵션 피일드 : type, address, month, note

proceedings conference proceeding을 통째로 인용할 때

필수 피일드 : title, year

옵션 피일드 : editor, volume 또는 number, series, address(conference 개최 장소), month, organization, publisher, note

techreport Technical Report

필수 피일드 : author, title, institution, year

옵션 피일드 : type, number, address, month, note

unpublished 출판되지 않은 자료

필수 피일드 : author, title, note

옵션 피일드 : month, year

### 13.3 Field

address 출판사 주소

annote

author 저자 이름

booktitle 책 이름

chapter

edition

editor

howpublished

institution

journal

key

month 책이 출판된 달  
 note  
 number 저널의 번호  
 organization conference sponsor organization  
 pages 인용된 페이지 범위, ex) 12-44, 67, 85-99  
 publisher  
 school 학위 논문이 쓰여진 학교  
 series 책의 series 이름  
 title  
 type technical report의 종류, ex) Research Note  
 volume 저널 volume  
 year 출판 연도

## 13.4 텍스트에 reference 달기

`\cite` 명령을 사용하여 텍스트에 reference를 표시한다.

```
\cite{reference-key, reference-key, ... }
```

```
\cite[page-number]{reference-key, reference-key, ... }
```

This book is made by Y. S. Choi and M. J. Kim. Main reference is `\LaTeX`for [2]. Especially the bibliography appears in the chapter 11[2, pp. 79-89] If you want information about `TeX`, You'd find it otherreference. [5, 7]

```

This book is made by Y. S. Choi
and M. J. Kim. Main reference is
\LaTeX for [1] dot s \cite{latex}.
Especially the bibliography appears
in the chapter 11
\cite[pp. 79-89]{latex}
If you want information about \TeX,
You'd find it otherreference.
\cite{latex, btxdoc}

```

## 13.5 BibTEX 을 이용해 bibliography 만들기

BibTEX 을 이용한 Bibliography 작성에 대하여 설명하겠다. Bibliography를 만들기 위해서는 다음의 두 명령을 .tex 화일에 포함시켜야 한다.

```
0 \bi bl i ogr aphy{mybi b}
```

이 명령은 bibliography를 만드는 명령으로 bibliography가 나타나기 원하는 위치에 이 명령을 삽입한다. mybib는 bibliography database 화일 (mybib.bib)의 이름이다.

```
0 \bi bl i ogr aphy st yl e{st yl e- opt i on}
```

Bibliography의 형식을 지정하는 명령으로 \begi n{document} 다음에 위치시킨다. 예)

```
\begi n{document}
```

```
\bi bl i ogr aphy st yl e{pl ai n}
```

0 사용가능한 style-option들은 다음과 같다.

**plain** 저자명, 연도, 제목으로 bibliography를 sort 한다. 일련 번호를 사용하여 refernce를 붙인다. 예) [3].

**unsrc** 기본적으로 plain과 같고 sort 하지 않는다.

**alpha** 기본적으로 plain과 같고 일련 번호 대신 저자명과 연도의 조합으로 reference를 붙인다. 예) [Knu86]

**abbrv** 기본적으로 plain과 같으며, 이름, 달, 저널이름에 약자가 사용된다.

0 이제 다음과 같은 순서로 bibliography를 만든다. Document는 mydoc.tex으로, bibliography database 화일은 mybib.bib로 가정한다.

1. L<sup>A</sup>T<sub>E</sub>X을 수행시킨다(l at ex mydoc). 이것으로 mydoc.aux 화일에 reference 한 문헌의 key가 기록된다.

2. BibTEX을 수행시킨다(`bi bt ex mydoc`). 이것은 `mydoc.aux` 파일을 읽고 `mybib.bib` 파일에서 문헌 정보를 찾아 `mydoc.bbl` 파일에 기록한다.
3. L<sup>A</sup>T<sub>E</sub>X을 다시 수행시킨다. 이것으로 Bibliography가 document에 포함된다.
4. L<sup>A</sup>T<sub>E</sub>X을 다시 한번 수행시킨다. 이것으로 Reference가 처리된다.

## 13.6 BibTEX 없이 bibliography 만들기

reference가 많지 않을 경우에는 굳이 BibTEX을 사용하지 않아도 된다. 이 경우에는 `t hebi bl i ogr aphy`를 이용하면 된다. bibliography가 나타날 곳에 다음과 같이 써 주면 된다.

```
\begin{t hebi bl i ogr aphy}{9}
\bi bit en{j amet} J. P. Jamet, P. Lederer, and A. H. Mudden,
Phys. Rev. Lett. {\bf 48}, 442(1982).
\bi bit en{moud} A. H. Mudden, E. C. Svensson, and G. Shirane,
Phys. Rev. Lett. {\bf 49}, 557(1982).
\bi bit en{gesi} K. Gesi, J. Phys. Soc. Jpn. {\bf 51}, 701(1982).
\end{t hebi bl i ogr aphy}
```

이 때의 결과는

### Reference

- [1] J. P. Jamet, P. Lederer, and A. H. Mouden, Phys. Rev. Lett. 48, 442(1982).
- [2] A. H. Mouden, E. C. Svensson, and G. Shirane, Phys. Rev. Lett. 49, 557(1982).
- [3] K. Gesi, J. Phys. Soc. Jpn. 51, 701(1982).

주의할 것은 이경우에 `\bi bit em`을 쓴다는 것과 `\begin{t hebi bl i ogr aphy}` 뒤에 붙는 것이다. 이는 `item label`의 최대크기를 나타낸다. 9라고 쓴 것은 위의

보기에서는 한 글자(숫자 하나)만 [ ]에 들어 가기 때문이다. `\bibitem` 뒤에 나오는 것은 BibTeX의 Key와 같은 역할을 하는 것이다. 텍스트에 reference번호를 붙이기 위해서는 `\cite{key}`를 사용하면 된다. 보통은 reference번호는 thebibliography input 순서대로 번호를 붙여주지만 이를 다음과 같이 쓸 수도 있다.

```
\begin{thebibliography}{Mudden 82}
\bibitem[Jamet 82]{jarnet} J. P. Jamet, P. Lederer, and A. H. Mudden,
Phys. Rev. Lett. {\bf 48}, 442(1982).
\bibitem[Mudden 82]{moud} A. H. Mudden, E. C. Svensson, and G. Shiran
Phys. Rev. Lett. {\bf 49}, 557(1982).
\bibitem[Gesi 82]{gesi} K. Gesi, J. Phys. Soc. Jpn. {\bf 51}, 701(1982)
\end{thebibliography}
```

여기에서 {Mudden 82}는 이것이 가장 긴 label이 되기 때문이다. 이 때의 출력은

## Reference

[Jamet 82] J. P. Jamet, P. Lederer, and A. H. Mudden, Phys. Rev. Lett. 48, 442(1982).  
[Mudden 82] A. H. Mudden, E. C. Svensson, and G. Shirane, Phys. Rev. Lett. 49, 557(1982).  
[Gesi 82] K. Gesi, J. Phys. Soc. Jpn. 51, 701(1982).

이러한 input을 마친 후에 결과를 보려면 TeX을 두번 실행시켜야 한다.



# 14장

## Glossary 와 Index 만들기

### 14.1 Glossary

LaTeX에는 Glossary를 만드는 특별한 기능이 없다. Glossary를 만드는 쉬운 방법은 `chapter`나 `section` 명령을 사용하는 것이다. 각 glossary 항목은 `description environment`를 사용한다. Glossary 제목 페이지에 `chapter` 또는 `section` 번호가 매겨지는 것이 싫으면 다음과 같이 한다.

```
\chapter*{Glossary}
\addcontentsline{toc}{chapter}{Glossary}
\pagestyle{myheadings}
\markboth{Glossary}{Glossary}
```

`addcontents` 명령은 `glossary`가 차례에 나타나게 하기위함이고 `pagestyle`과 `markboth` 명령은 페이지 헤더에 `"Glossary"`가 나타나게 한다.

## 14.2 Index

$\LaTeX$ 은 `index`를 자동으로 생성하는 기능을 갖고 있다. 생성된 `index`는 텍스트 파일이지만, `Makeindex` 프로그램을 사용하면 `index`의 `formatting` 까지 자동으로 할 수 있다.

다음의 설명들은 `Makeindex`를 사용하는 방법을 설명한다. `index`의 생성은 다음의 두 가지 절차를 거친다.

1. `index`에 포함시키고자 하는 단어들을 `mark` 한다.
2. `Makeindex`를 수행시켜 `index`를 생성한다.

### 14.2.1 index 할 단어의 mark

0  $\LaTeX$ 은 `\index` 명령을 사용하여 `index`할 단어를 `mark`한다. 기본적인 `index` 명령의 형태는 다음의 예와 같다.

```
The Roll s Royce\index{Roll s Royce} is made in Engl and.
```

0 위와 같이 같은 단어를 두번 입력하는 것이 귀찮으면 다음과 같이 `macro`를 정의하여 `index`를 간편히 할 수 있다. `\newcommand`는 `document preamble`에 위치시켜야 한다.

```
\newcommand{\mi}[1]{#1\index{#1}}
```

`\mi`의 사용예

```
The \mi{Roll s Royce} is made in Engl and.
```

본절에서는 `\mi`가 위와 같이 정의된 것으로 간주한다.

0 문서에서 `\Rolls Royce`가 네번 `mark`되었다면, 다음과 같은 형태의 `index`가 만들어진다.

```
Rolls Royce, 3, 14, 76, 105
```

0 한 페이지에서 두번 mark된 경우에는, index에 그 페이지 번호가 한번만 나타난다.

0 Index에 sub-entry, sub-sub-entry, cross-reference도 만들수 있다. main-entry, sub-entry, sub-sub-entry 사이는 \!" 로 구분하고 cross-reference는 \|" 로 구분한다.

예)

```
Maserati, 50
Rollce Royce, 3, 14, 76, 105
    manufacturing techniques of, 77
    popularaty of, 5
sports cars, 13, 44
    racing, see Maserati
        corporate participation in, 51
```

입력:

```
pp 5: Roll s Royce\index{Roll s Royce! popul arity of}
pp 13: \mi {sports cars}
pp 44: \ni {sports cars}
pp 50: \mi {Maserati}
pp 50: sports cars\index{sports cars! raci ng| see{Maserati}}
pp 51: sports cars\index{sports cars! raci ng! corporat e parti ci pat i on i n}
pp 77: Roll s Royce\index{Roll s Royce! manuf act uri ng techni ques of}
```

주의 : sub-entry, sub-sub-entry나 cross-reference의 index를 할 때는 \mi 대신 \index 를 사용하여야 한다.

0 Index entry에 문서의 일정 range를 지정하기.

index range 시작 명령 : \index{ ... |{}

index range 끝 명령 : \index{ ... |)}

예)

Pusan, 86  
Hyundai Motor Co., 86  
    Excel, 86{ 92  
    Scoupe, 92{ 93  
Kia Motors Corp., 101-109

입력:

```
pp 86: \mi {Pusan}
pp 86: \mi {Hyundai Mbt or Co.}
pp 86: Hyundai Mbt or Co. \index{Hyundai Mbt or Co. ! Excel | (}
pp 92: Hyundai Mbt or Co. \index{Hyundai Mbt or Co. ! Excel |)}
pp 92: Hyundai Mbt or Co. \index{Hyundai Mbt or Co. ! Scoupe | (}
pp 93: Hyundai Mbt or Co. \index{Hyundai Mbt or Co. ! Scoupe |)}
pp 101: Ki a Mbt or s Corp. \index{Ki a Mbt or s Corp. | (}
pp 109: Ki a Mbt or s Corp. \index{Ki a Mbt or s Corp. |)}
```

## 0 주의 사항

{ index keyword 앞뒤에 불 필요한 space를 넣지 말것. 예를 들어, 다음의 index marker는 각기 다른 index entry를 생성한다.

```
\index{Must ang}
\index{ Must ang}
\index{Must ang }
```

{ index keyword는 대소문자를 구별한다.

{ index marker의 위치는 section heading 바로 밑이나 문단의 시작이나 끝이 좋다. 문단 중의 단어 사이에 index mark를 하면 그 단어 사이가 조금 벌어지게 된다.

## 14.2.2 MakeIndex를 사용한 index의 생성

0 MakeIndex를 수행하기 전에

```
{ \documentstyle에 makeidx 옵션 추가. 예,  
  \documentstyle[11pt,makeidx]{book}  
  
{ \documentstyle 바로 다음에  
  \makeindex  
  
{ \end{document} 바로 이전에  
  \cleardoublepage  
  \printindex
```

0 Index 생성

1. 일단  $\text{\LaTeX}$ 을 수행시켜 `.idx` 파일을 만든다.
2. 만들어진 `.idx` 파일이 `my-le.idx` 라고 가정하면, 다음의 명령으로 `my-le.ind` 파일을 생성한다.  

```
makeindex myfile.idx
```
3. 이제  $\text{\LaTeX}$ 으로 `my-le.tex`을 다시 한번 수행하면 문서의 끝에 `index`가 만들어진다.



# 15장

## Two-Column Document

### 15.1 Twocolumn 옵션

`twocolumn` 옵션을 사용하면 `document` 전체를 `two-column`으로 포맷할 수 있다.  
예)

```
\documentstyle[twocolumn]{article}
```

### 15.2 Twocolumn Proceeding 옵션

- 0 Conference editor 중 상당수는 Letter 크기의 종이에 laser-printer로 찍은 논문을 접수한다.  $\text{\LaTeX}$ 은 ACM과 IEEE conference proceedings 규격에 맞는 출력을 생성하는 "proc" style을 제공한다.
- 0 `proc style` 옵션은 `article document style`에서 동작하도록 설계되어 있다.  
예)

```
\documentstyle[proc]{article}
```

- 0 `proc` 옵션을 사용할 때는 `\begin{document}` 바로 뒤에서 `\copyright space` 명령을 위치시켜야 한다. 이것은 공란의 `footnote`를 첫 페이지 첫 `column` 아래에 만드는 것으로, `proceedings editor`가 `copyright notice`를 쓸 자리를 비워두는 것이다.
- 0 만약 첫번째 페이지 첫 번째 `column`에서 `\footnote` 명령을 사용하고자 하면, `\copyright space` 명령을 `footnote` 명령 다음에 오도록 해야 한다.
- 0 매 페이지 하단에 저자명과 제목을 표시하고 싶으면 다음의 명령을 `\begin{document}` 전에 `preamble`에 위치시킨다.
 

```
\markright{Name---Header Title}
```

### 15.3 텍스트 중에서 two-column으로 전환

0 `\twocolumn`

다음 페이지로 넘어가서 `two-column`으로 포매팅을 시작한다.

0 `\onecolumn`

다음 페이지로 넘어가서 `one-column`으로 포매팅을 시작한다.

`two-column`으로 포매팅하는 중에 `\newpage` 명령을 수행하면 다음 페이지로 넘어가는 대신 다음 `column`으로 넘어간다.

### 15.4 Minipage environment

$\text{\LaTeX}$ 은 원하는 폭의 가상 페이지(`minipage`)를 만들고 그 페이지 안에서 포매팅하는 기능을 제공한다. 또한 포매팅된 `minipage`들을 각각 사각 상자 모양의 문자처럼 간주하여 포매팅할 수도 있다. `minipage` 명령을 사용한 예로는 `figure`, `table environment`가 있다.

```
\begin{minipage}{page-width}
```

페이지 내용



`\end{mini page}`

`\begin{mini page}[t or b]{page-width}`

페이지 내용

`\end{mini page}`

0 `page-width` : 가상 페이지의 폭

0 포매팅이 끝난 `mini` 페이지는 하나의 문자로 간주되어 포매팅 되는데, 이때 `mini` 페이지의 중심을 `baseline`에 맞춘다. 옵션 `t` 또는 `b` 를 쓰면 `mini` 페이지의 중심을 위쪽 끝 또는 아래 쪽 끝으로 맞출 수 있다.

예)

<code>This is the -rst mini page.</code>	<code>This is the second mini page. This is the second mini page.</code>	<code>This is the third mini page. This is the third mini page. This is the third mini page.</code>
--	--	---

입력:

`\begin{mini page}{1.2in}`

`This is the first mini page.`

`\end{mini page}`

---

`\begin{mini page}{1.2in}`

`This is the second mini page.`

`This is the second mini page.`

`\end{mini page}`

---

`\begin{mini page}{1.2in}`

`This is the third mini page.`

`This is the third mini page.`

`This is the third mini page.`

```
\end{mini page}
```

예)

```
This is the -rst mini | This is the second | This is the third mini
page.                  mini page. This is the  page. This is the
                        second mini page.  third mini page. This
                        is the third mini page.
```

입력:

```
\begin{mini page}[t]{1.2in}
```

```
This is the first mini page.
```

```
\end{mini page}
```

---

```
\begin{mini page}[t]{1.2in}
```

```
This is the second mini page.
```

```
This is the second mini page.
```

```
\end{mini page}
```

---

```
\begin{mini page}[t]{1.2in}
```

```
This is the third mini page.
```

```
This is the third mini page.
```

```
This is the third mini page.
```

```
\end{mini page}
```

예)

```
미니 페이지a 에서 footnote를 사용
footnote in 할 수 있습니다. mini page.
```

<sup>a</sup> 포맷팅되는 가상 페이지

입력:

```
footnote in
```

```
\begin{mini page}{2in}
```

미니 페이지 `\footnote{` 문자처럼 포매팅되는 가상 페이지

에서 `\footnote`를 사용할 수 있습니다.

`\end{mini page}`

`mini page.`



## 16장

# 여러가지 유용한 스타일 화 일

이장에서는 한글  $\text{\LaTeX}$ 의 기본 명령어외에 제공되는 여러가지 유용한 매크로 (`*.tex`) 또는 또는 스타일화일(`*.sty`) 에 대하여 그 쓰임 예를 보이겠다. 이 명령어또는 환경을 쓰고자하면 필히 그 환경을 정의하는 매크로 화일을 다음과 같이 `preamble`에 표시하여야 한다.

예로서 `multi cols.sty`의 환경을 이용하려면:

```
\documentstyle[multi cols]{article}
```

로서 환경정의를 하여주고 `multi cols.sty`의 환경에서 제공하는 명령어를 쓸수있다.

## 16.1 multicol.sty

`twocol um. sty`에서는 `column`의 수를 변환 할때 새로운 페이지가 시작되나, 이 `multicol s. sty`에서는 한 페이지 안에서 `column`의 수를 바꿀수 있다. 다음의 예를 보시오.

```
\begin{multicols}{3}
```

이 세상에서 가장 슬프고 불쌍한 것은 외로운 자에게 사랑의 불씨가 날아와서 다시 머나먼 하늘나라로 날아가 버리는 것임을 우린 잘 알고 있기에 또 다시가 없는 인생 속에서 다시 사랑하는 그를 찾아 이렇게 작은 인생길을 걸어 갑니다. 세상은 예전과 같지 않기에 이런 이야기를하기에는 너무나 퇴폐했고 무관심해 졌습니다. 참으로 뼈아픈 현실이 아닐 수 없습니다. 세상이 처음 열리고 나의 반쪽의 대상을 처음 만난 사람, 그 감정은 안으로만 사무치어 마음에서 불러보고 왜라는 의문을 하면서 스스로 나의 사랑을 저버리고 가는 바보가 됩니다. 언젠가 그는 사랑은 믿음이라고, 훌륭한 일은 인내로 하는 것이라고 말했을 때 도무지 무슨 의미인지 몰랐지만 지금은 알 것 같습니다.

```
\end{multicols}
```

이 세상에서 가장 슬프고 불쌍한 것은 외로운 자에게 사랑의 불씨가 날아와서 다시 머나먼 하늘나라로 날아가 버리는 것임을 우린 잘 알고 있기에 또 다시가 없는 인생 속에서 다시 사랑하는 그를 찾아 이렇게 작은 인생길을 걸어 갑니다. 세상은 예전과 같지 않기에 이런 이야기를하기에는 너무나 퇴폐했고 무관심해 졌습니다. 참으로 뼈아픈 현실이 아닐 수 없습니다. 세상이 처음 열리고 나의 반쪽의 대상을 처음 만난 사람, 그 감정은 안으로만 사무치어 마음에서 불러보고 왜라는 의문을 하면서 스스로 나의 사랑을 저버리고 가는 바보가 됩니다. 언젠가 그는 사랑은 믿음이라고, 훌륭한 일은 인내로 하는 것이라고 말했을 때 도무지 무슨 의미인지 몰랐지만 지금은 알 것 같습니다.

## 16.2 fancybox.sty

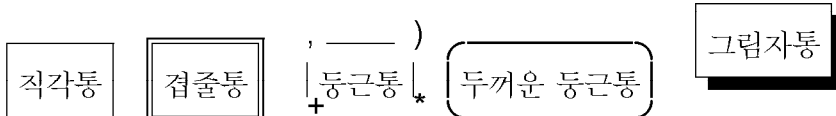
`fancybox.sty`는 Timothy Van Zandt (`tvz@Princeton.EDU`)가 만든것으로서 글자를 여러가지 통안에 표시하는 매크로를 제공한다.

다음 보기를 보시오.

```
\fbox{\strut 직각통} \doublebox{\strut 겹줄통}
```

```
\ovalbox{\strut 둥근통} \Ovalbox{\strut 두꺼운 둥근통}
```

```
\shadowbox{\strut 그림자통}
```



### 16.3 verbatim.sty

많은 경우에 `verbatim` 환경안에 화일내용을 직접 넣지않고 간접적으로 그화일 이름을 지시함으로 `LaTeX` 가 처리한다. (예: 여러가지 프로그램화일을 `LaTeX` 포함시킬때.)

이외에도 `verbatim` 환경에 포함될 수있는 텍스트의 크기가 제한되어 있으므로 `verbatim.sty`을 `\documentstyle` 의 option에 포함시키는것이 중요하다.

다음의 보기를 활용하십시오.

```
\begin{quote}\small
\verbatiminput{hello.c}
\end{quote}
```

그 출력은

```
#include <stdio.h>

main(argc, argv)
int argc;
char *argv[];
{
    int i, iter;
    float ctme, total_ops;
    switch (argc) {
        case 2:
            iter = atoi(argv[1]); /* Get the number of iteration! */
            break;
```

```

    default:
printf("default iter=1\n");
iter = 1;
};
for(i=0;i<iter;i++)
    printf(" <%d> Hello World!\n",i);
}

```

이 `verbatim` 환경을 이용하여 임의의 텍스트 파일을 찍을수있는 DOS의 BATCH 파일을 다음과 같이 쓸수 있다.

```

REM
REM Printing Text file using Latex...
REM
echo \documentstyle[fancyheadings,fullpage]{article} > tmp.hlx
echo \begin{document} >> tmp.hlx
echo \pagestyle{fancy} >> tmp.hlx
echo \lhead{%d} >> tmp.hlx
echo \rhead{\bf\the page} >> tmp.hlx
echo \footnoteseize >> tmp.hlx
echo \begin{verbatim} >> tmp.hlx
type %d %2 %3 >> tmp.hlx
echo \end{verbatim} >> tmp.hlx
echo \end{document} >> tmp.hlx
hlx2tex tmp
virtext &hwl at ex tmp

```



# 17장

## 특수 기능

### 17.1 Newcommand, Renewcommand를 사용한 Macro의 생성

0 `\newcommand` 명령으로 자주 쓰는 텍스트나 명령들을 하나의 `macro` 명령으로 정의할 수 있다.

```
\newcommand{\macro 이름}{macro 내용}
```

```
\newcommand{\macro 이름}[argument 갯수]{macro 내용}
```

정의 내용에서 `argument` 는 # 1, # 2, ... , # 9 로 표시한다. 따라서 최대 9개 까지의 `argument` 를 사용할 수 있다. 예를 들면,

1. `\newcommand{\ja}{\it Journal of Algorithm/}`

`\ja` 를 쓰면 그 위치에 `Journal of Algorithm`이 나타난다.

2. `\newcommand{\bi}{\begin{it emi ze}}`

```
\newcommand{\ei}{\end{it emi ze}}
```

3. `\newcommand{\al}{\$al pha$}`

```
\newcommand{\be}{\bet a$}
```

```
\newcommand{\ga}{\gamma$}
```

0 Argument를 가지는 매크로도 정의할 수 있다. 예를 들면,

```
1. \newcommand{\lplace}[1]{\frac{\partial ^2 #1}{\partial x^2}
+ \frac{\partial ^2 #1}{\partial y^2}$}
```

```
\lplace{A} !  $\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2}$ 
```

```
2. \newcommand{\mean}[2]{\frac{#1 + #2}{2}$}
```

```
\mean{A}{B} !  $\frac{A+B}{2}$ 
```

0 \renewcommand 명령을 사용하면 이미 정의된 macro를 재정의할 수 있다. 예를 들면,

```
\renewcommand{\bf}{\it}
```

document의 \bf가 \it 처럼 동작한다. \renewcommand는 document preamble에 위치시켜야 한다.

## 17.2 Newenvironment, Newtheorem

0 \newenvironment 명령으로 새로운 environment를 정의할 수 있다.

```
\newenvironment{name}{침 명령} 끝 명령}
```

예)

```
\newenvironment{nypoem}{\begin{verse} \end{verse}}
```

```
\begin{nypoem}
```

```
Reason, \\\
```

```
when used for discovery\\
```

```
rather than defense, \\\
```

```

is an intellectually honest enterprise\
and hence one of life's greatest pleasures.
\end{mypoem}

```

- 0 수학적 서술을 할 때는 theorem, proposition, lemma, conjecture등을 사용하고 이들에게 일련 번호를 매긴다. \newtheorem 명령은 이러한 theorem, proposition, 등에 일련 번호를 붙이는 environment를 정의한다.

```
\newtheorem[theorem environment - name]{theorem class - name}
```

theorem-class-name : theorem, proposition, lemma,

```
\newtheorem[myth]{Theorem}
```

```
.
.
.
```

```
\begin{myth}
```

This is first theorem

```
\end{myth}
```

```
\begin{myth}
```

This is second theorem

```
\end{myth}
```

위의 예에서는 theorem에 document 전체적으로 일련 번호가 매겨진다.

- 0 chapter나 section 내에서 따로 일련 번호를 매길 수도 있다.

```
\newtheorem[theorem environment - name]{theorem class - name}[chapter or section]
```

```
\newtheorem{myprop}{Proposition}[chapter]
```

```
.
.
.
```

```
\begin{myprop}
```

This is the first proposition in Chapter 7.

```
\end{myprop}
```

Proposition 7.1 This is the -rst proposition in Chapter 7.

## 17.3 Counter

$\LaTeX$ 에서 `chapter`, `section` 과 같이 번호가 자동적으로 매겨지는 것들은 그 번호를 기억하기 위해 `counter`를 사용한다. 표 17.1는  $\LaTeX$ 에서 사용하는 `counter`들이다. `\set counter` 명령을 사용하여 `counter` 값을 바꿀 수 있다.

```
\set counter { counter - name } { new - val ue }
```

```
0 \set counter { chapter } { 7 }
```

현재 `chapter` 번호를 7로 바꾼다. 따라서 다음의 `\chapter` 명령은 `chapter 8`을 표시한다.

```
0 \set counter { page } { 74 }
```

현재의 페이지 번호를 74로 바꾼다.

```
0 \addto counter { equation } { 4 }
```

`Counter`에 주어진 값을 더한다.

0  $\LaTeX$ 에서 숫자는 다양한 형태로 표현될 수 있다.

예를 들어, `page counter`가 7일 때,

명령	결과
<code>\arabic{page}</code>	7
<code>\Roman{page}</code>	VII
<code>\roman{page}</code>	vii
<code>\Alph{page}</code>	G
<code>\alph{page}</code>	g

0  $\LaTeX$ 에서 `counter` 값을 표시하는데 사용하는 `macro`는 `\the`에 `counter-name`을 붙인 형태를 가진다. 예를 들어, 페이지 번호를 표시하는 `macro`는 `158`이다.

`\the page` `macro`를 재정의 함으로써 페이지 번호의 표시 형태를 바꿀 수 있다. 다음 예는 보통의 페이지 번호 대신 `chapter` 번호와 `chapter` 내에서의 페이지 번호를 `document`의 페이지 번호로 사용하는 예이다. 즉, 페이지 번호가 1-1, 1-2, 1-3, 2-1, 2-2, 2-3, ... 으로 표시된다.

```
\renewcommand{\the page}{\arabic{chapter}--\arabic{page}}
\newcommand{\mychapter}[1]{\chapter{#1}\setcounter{page}{1}}
```

`\renewcommand` 명령은 `document preamble`에 위치시켜야 한다. 새로운 `chapter`가 시작될 때 마다 `\mychapter` 를 사용한다.

0 다음과 같이 새로운 `counter`를 만들수도 있다.

```
\newcounter{new-counter-name}
```

도표 17.1:  $\text{LaTeX}$ 에서 사용하는 section 번호와 그 counter

Unit Name	Counter Name
chapter	chapter
enumerate 1st level	enumi
enumerate 2st level	enumii
enumerate 3st level	enumiii
enumerate 4st level	enumiv
equation	equation
- gure	- gure
footnote	footnote
minipage footnote	mpfootnote
page	page
paragraph	paragraph
part	part
section	section
subparagraph	subparagraph
subsection	subsection
subsubsection	subsubsection
table	table

# 18장

## Error

### 18.1 자주 발생하는 에러

TEX에서 에러가 잘 생기는 곳은 다음과 같다.

1. 특수기호를 그냥 입력한 경우. # \$ % & \_ { } 등이 제 위치에 적절히 오지 않는 경우가 많다.
2. f와 g가 잘 맞지 않는 경우.
3. 명령어나 environment를 틀리게 입력한 경우.
4. section, chapter, caption등을 너무 길게 쓰는 경우.
5. 수학모드에서만 써야하는 명령어를 일반 문장에서 사용하는 경우.
6. 수학모드에서 괄호를 제대로 사용하지 않을 경우.
7. document style, \begin{document}, \end{document}가 빠졌을 경우.

## 18.2 에러를 어떻게 고칠 것인가

TEX을 실행시키면 이상한 메시지들이 많이 나온다. 이를 보고 어디가 잘못 되어서 에러가 발생했는지를 찾는다는 것은 쉬운 일이 아니다. 이를 쉽게 찾기 위해서는 숙련이 필요하지만 최소한 어느 정도는 할 수 있어야 한다. 에러가 나올 때는 다음과 같이 화면에 나타날 것이다.

```
... (myfile.tex
! Undefined control sequence.
l.6 \begin{enmr at e}
?
```

? 위에 `l.6 \begin{enmr at e}` 가 보인다. 이 경우는 입력 -le의 6번째 줄에 error가 생긴것이다. `l.6`가 `\line 6`를 뜻한다. ! 뒤에 나오는 내용이 틀린 이유를 설명해주고 있다. 이 경우는 `enumerate`를 `enumrate`로 잘못 입력함으로써 생긴 것이다. `\?`가 나왔을 때 작업을 중단하고 나오려면 `x`를 치고 **Enter**, 계속 다른 어떤 에러가 있는지 보려면 `s`나 **Enter**를 치면 된다.

## 18.3 경고

이는 출력이 나오는 데에는 영향이 없으나 최종출력이 원하는 대로 나오지 않을 수도 있다는 것을 뜻한다. 주로 지적해주는 내용은 `textwidth`의 초과, 인용이나 `cross-reference`에서의 착오 등이다. `textwidth`의 초과는 `overflow` 이라고 경고가 나온다. 인용이나 `cross-reference`의 경고는 출력결과가 그위치에서 `\?`로 표시된다. 여기에서 주의할 점은 `cross-reference`, `\ref`, `\cite`등을 사용한 경우는 처음 TEX을 실행시키면 반드시 경고가 나온다. 이는 TEX을 한번 더 실행시키면 없어진다. 두번 실행시켜도 없어지지 않으면 지적된 부분을 고쳐야한다. 이들은 모두 `.log -le`에 기록 되어 있다.



## 제 III 부

# 보기화일 Samples



# 19장

## Samples 화일

본 장에는 한글 WinT<sub>E</sub>X에 포함된 보기화일의 일부를 리스팅하였다. 필히 한글 WinT<sub>E</sub>X에디터를 이용하여 입력시켜 보시고 T<sub>E</sub>X 컴파일하여 프린팅해보십시오.

### 19.1 바자회 알림

```
\documentstyle[12pt,fullpage,fancybox]{article}
\begin{document}
\pagestyle{empty}
\begin{center}
  \shadowbox{\Huge\gs 대흥 바자회 알림}
\end{center}

\large\dr
문안드립니다.
Long Beach에 위치한 대흥에서 다음과 같이 바자회가 열리오니
다음과 같이 마켓 제품 보다 질 좋고 싱싱한 제품을 싸게 구입하시고
즐거운 교제의 시간을 보내시기 바랍니다.

\gt
\begin{itemize}
%\item {\tt 일시 :} 1994년 3월 12일 토요일 오후 2:00-6:00

\item {\tt 일시 :} 1994년 3월 12일 토요일 오후 2:00-6:00
\hspace{0.5in}\special{!scale co, ee.wmf, 30mm 40mm}

\item {\tt 장소 :} 대흥 주차장 \\\
  {\tt 1401 강남구 신사동 \\\
  대한민국 \\\
  310)424-1994}
\end{itemize}
\dr
$\bullet$ 품목및 가격 \\\

\gt
```

```

\begin{tabular}{| r | r | }
\hline\hline
고추장 & \$15.00/ gal & 도라지 & \$5.00/ lb \\
된장 & \$12.00/ gal & 고사리 & \$4.50/ lb \\
간장,국간장 & & 미역(아씨표 차자) & \$4.50 \\
조기(특,참조기) & \$2.00/lb & 새우젓 & \$15.00/ gal \\
김 (오양,구운것) & \$1.70/6sheets & 이면수 & \$2.5/ pack \\
굵은멸치 & \$6.00/lb & 갈비살 & \$1.50/lb \\
오징어채 & \$9.00/lb & 돼지고기 (살코기) & \$1.70/lb \\
표고버섯 & \$6.00/lb & 사태살 & \$2.00/lb \\
고구마 & \$0.60/lb & 청포묵 & \$1.50/ pack \\
참깨 & \$1.00/lb & 빈대떡 & \$2.50/ pack \\
취나물 & \$8.00/lb & & & 김밥 & \$2.50/2roll \\
오뎅국 & \$1.50 & 식혜 & \$3.00/ half gallon \\
\hline\hline
\end{tabular}

\rm
\begin{itemize}
\item 오시는 모든 분들께 Lemon Tea, Cool Drink 는 FREE로 드립니다.
위의 품목 외에도 여러 품목이 준비되어 있습니다.

\item 미리 예약을 원하시거나 문의가 있으시면 한성례 회장 (310)549-4647나
김옥영 총무 310)220-2421 에게 해주십시오.
\end{itemize}
\end{document}

```

## 19.2 동창회 정관

```

\documentstyle[multicols,fullpage,11pt]{article}
\begin{document}
\pagestyle{empty}

\def\theenumii{\han{\value{enumii}}항}
\def\p@enumii{\theenumii\theenumii}
\def\labelenumii{\theenumii}
\def\labelenumi{\gs\theenumi 조}

\begin{center}
{\Huge\gs 동산 동창회 정관} \\
\large 1994년 2월 12일 \\
\end{center}
\parskip= 0.5em

\begin{multicols}{2}
\begin{enumerate}
\begin{center}
{\large 제 1 장: 총 칙} \\
\end{center}
\small\dr
\item 명칭 : 본회는 명칭은동산학교 동창회라 칭한다.
\item 장소 : 본회는 동산학교안에 둔다.
\item 목적 :
본회는 회원간의 친목을 도모하고 동산 학교의 부흥과 발전에 기여한다.
\begin{center}

```

```

{\large 제 2 장: 회원 }
\end{center}
\item 본회의 회원은 본학교에 졸업한 사람으로 기혼 또는 만 30세 이상의 남자로서 한다.
\item 회원의 의무와 권리 :
\begin{enumerate}
\item 본회 회원은 제 3조의 목적을 달성하기 위하여 제반 사업 및 행사에 참여할 권리와 의무가 있다.
\item 본회 회원은 회칙 및 본회에서 결의된 사항을 준수하여야 한다.
\item 본회 회원은 선거권, 피 선거권, 발언권 및 결의권이 있다.
\item 본회 회원은 소정의 회비를 납부할 의무가 있다.
\end{enumerate}

```

```

\begin{center}
{\large 제 3 장: 조직 }
\end{center}
\item { 임원 및 부서 }
\begin{enumerate}
\item \label{manorg} 임원의 구성은 회장 1명, 부회장 1명, 서기 1명 회계 1명으로 한다.
\item 부서는 친교부, 봉사부 행사부로 구성되며 각부는 부장 1인 및 부원으로 구성된다.
\end{enumerate}

```

```

\item { 임무 }
\begin{enumerate}
\item 회장은 본회를 대표하며 본회 운영에 대한 제반 사항을 감독 및 통솔한다.
\item 부회장은 회장은 보필하며 회장 유고시 이를 대행한다.
\item 서기는 본회 제반 기록과 문서를 관장한다.
\item 회계는 본회의 회비 징수 및 지출을 하며 재정에 대한 일체를 담당한다.
\item 대외부는 대외적으로 활동을 하며 대내적으로 기도원 또는 야외 수련등을 통하여 애교심을 증진 시킨다.
\item 친교부는 새회원 영접 및 회원간의 친목을 위한 제반사항을 담당한다.
\item 봉사부는 학교 또는 회원중에 도움을 필요로 하는 일이있을때 솔선적으로 참여한다.
\item 행사부는 회원간의 친목을 도모하기 위한 건전한 운동을 계획 실행한다.
\end{enumerate}

```

```

\begin{center}
{\large 제 4 장: 회의 }
\end{center}
\item 총회 : 매년 1회 즉 12월에 소집하여 임원 개선과 중요안건을 토의하며 회칙수정 및 결산 보고와 새해예산을 정한다.
\item 임시총회 : 회원의 과반수사 요구할때 회장이 소집한다.
\item 월례회 : 매월 1회 정기소집 하며 당월 진행사항 보고 및 다음달 행사등을 토의한다.
\item 임원회 : 회장이 필요할때 소집하되 1주일 전에 장소와 일시를 공고한다.
\item 성원 및 결의: 본회의 모든 회의는 참여회원의 성원으로하며, 참여회원의 2/3이상의 찬성으로 한다.
\item 선거 : 모든 임원 (\ref{manorg}) 은 총회에서 투표로 선출하며 그 투표방법은 총회에서 결정한다.
\item 보선 : 임원중에 결원이 생길 경우에는 임시총회를 소집하여 이를 보선한다.
\item 임기 : 모든 임원의 임기는 1년으로 하며 보선의 임기는 잔여기간으로 한다.

```

```

\begin{center}
{\large 제 5 장: 재정 }
\end{center}
\item 회비 : 본회의 재정은 회원의 회비 및 찬조금으로 하며, 회비는 회원당 매월 ₩10.00이상으로 한다.
\item 회계년도 : 본회의 사업 회계년도는 당년 1월 부터 12월로 한다.

```

```

\begin{center}
{\large 제 6 장: 부칙 }
\end{center}

```

```

\item[] 본회의 회칙은 통과후 즉시 효력을 발생한다.
\end{enumerate}
\end{multicols}
\end{document}

```

## 19.3 수학기호

```

\documentstyle[11pt]{article}
\begin{document}
\begin{center}
{\Huge\mj 산수기호}
\end{center}

$$\prod_{j \ge 0} \biggl( \sum_{k \ge 0} a_{jk} z^k \biggr)
= \sum_{n \ge 0} z^n \cdot \biggl( \sum_{\substack{\scriptstyle k_0, k_1, \dots \ge 0 \\ \scriptstyle k_0 + k_1 + \dots = n}} a_{0k_0} a_{1k_1} \dots \biggr) .$$

$$\binom{n_1 + n_2 + \dots + n_m}{n_1, n_2, \dots, n_m}
= \binom{n_1 + n_2}{n_1} \binom{n_1 + n_2 + n_3}{n_2} \dots \binom{n_1 + n_2 + \dots + n_m}{n_m} .$$

$$\def\# 1# 2{(1-q^{# 1 # 2+n})} % to save typing
\prod_{n=0}^{\infty} \frac{1-a_1 a_2 \dots a_n}{1-b_1 b_2 \dots b_n} .$$

$$\underbrace{\overbrace{\mathstrut a, \dots, a}^{k; a \mathchar" \rm s}}_{\overbrace{\mathstrut b, \dots, b}^{l; b \mathchar" \rm s}} \{k+ l \mathchar" \rm elements\} .$$

$$\pmatrix{ a & b \cr c & d \cr } & \pmatrix{ e & f \cr g & h \cr } \cr
\smallskip
0 & \pmatrix{ i & j \cr k & l \cr } \cr .$$

$$\det \left| \begin{matrix} c_0 & c_1 & \dots & c_n \\ c_1 & c_2 & \dots & c_{n+1} \\ c_2 & c_3 & \dots & c_{n+2} \\ \vdots & \vdots & \ddots & \vdots \\ c_n & c_{n+1} & \dots & c_{2n} \end{matrix} \right| > 0 .$$

\end{document}

```

## 19.4 논문

```

\documentstyle[11pt]{harticle}
\begin{document}
\title{\huge 詩選}
\author{DuckHyun Kim}
\maketitle

```

\tableofcontents

\section{가보지 못한 길}

{\it 노랗게 This is English..물든 숲속에 두갈래 길이 있습니다}

한 나그네 몸으로 두길을 다가볼 수 없어, 아쉬운 마음으로 그곳에 서서  
한쪽 길이 덩불속으로 감돌아간 끝까지 한참을 그렇게 바라보았습니다.  
그리고는 다른쪽 길을 택했습니다.  
먼저길에 못지 않게 아름답고 어찌면 더 나은 듯도 싶었습니다.  
사람들의 밝은 흔적은 비슷했지만 풀이 더 무성하고 사람의 발길을 기다리는 듯  
해서였습니다.

먼저 길은 다음으로 생각했습니다.  
같은 길로 이어지는 것이기에 다시 돌아오기 어려우리라 알고 있었지만  
먼 훗날 어디에선가 나는 한숨쉬며 이야기를 할 것입니다.  
숲속에 두갈래 길이 있어 나는 사람이 덜 다닌 길을 택했습니다.  
그리고 내 인생을 이처럼 바꿔 놓은 것입니다. 라고...

\subsection{서정주 : 푸르른날}

\begin{center}\it

눈이 부시게 푸르른 날은  
그리운 사람을 그리워하자.

저기 저기 저 가을 꽃 자리  
초록이 지쳐 단풍드는데

눈이 내리면 어이하리야,  
봄이 또 오면 어이하리야.

내가 죽고서 네가 산다면 !  
네가 죽고서 내가 산다면 !

눈이 부시게 푸르른 날은  
그리운 사람을 그리워하자.  
\end{center}

\subsection{윤동주: 서시}

죽는 날까지 하늘을 우러러  
한 점 부끄럼이 없기를  
잎새에 이는 바람에도  
나는 괴로와 했다.  
별을 노래하는 마음으로  
모든 죽어 가는 것을 사랑해야지  
그리고 나한테 주어진 길을  
걸어 가야겠다.  
오늘 밤에도 별이 바람에 스치운다.

\section{알수 없어요}

바람도 없는 공중에 수직의 파문을 내며, 고요히 떨어지는  
오동잎은 누구의 발자취입니까?

지리한 장마끝에 서풍에 물려가는 무서운 검은 구름의 터진  
틈으로, 언뜻언뜻 보이는 푸른 하늘은 누구의 얼굴입니까?

꽃도 없는 깊은 나무에 푸른 이끼를 거쳐서, 예 탐 위의  
고요한 하늘을 스치는 알 수 없는 향기는 누구의 입김입니까?

근원은 알지도 못할 곳에서 나서 돌부리를 울리고, 가늘게  
흐르는 작은 시내는 굵이굵이 누구의 노래입니까?

연꽃 같은 발꿈치로 가이 없는 바다를 밟고, 옥 같은 손으로  
끝없는 하늘을 만지면서, 떨어지는 해를 곱게 단장하는 저녁놀은  
누구의 시입니까?

타고 남은 재가 다시 기름이 됩니다.  
그칠 줄을 모르고 타는 나의 가슴은 누구의 밤을 지키는 약한  
등불입니까?

```
\end{document}
```

## 19.5 책소개

```
\documentstyle[multicols,fancybox,fullpage,11pt]{article}
\font\bigbiggs=hgs scaled 5600
\begin{document}
\hspace{0.0in} \special{isoscale books.wmf, 40mm 40mm}
\hspace{4.5in} \special{isoscale books.wmf, 40mm 40mm} \
\begin{center}
%\shadowbox{\bigbiggs\char180\char235\char200\char239\char195\char165\char188\char210\char
%\bigbiggs 대흥 책소식}\
{\Huge\gs 대흥 책소식}\
\vskip 1em
\gs 1994년 2월 서적 신간안내 \
\tt 1234 Western St. Los Angeles, CA 987654 \
{\hH\char162\char207\ } 012)234-5678 \ovalbox{\small FAX} 123)456-7890
\end{center}

\newcommand{\book}[4]{
\noindent{
\parbox{\hsize}{\hH\char162\char192}{\tt # 1}}\
\noindent \mbox{\ } \h-lll {\large\gs (# 2)}\
%\large\gs (# 2) \
\rm # 3 (\$/# 4)}
\vskip 0.5em}

\newcommand{\bsect}[1]{
\vskip 0.5em
\noindent \rule[0.05in]{\hsize}{0.02in} \
\centerline{\LARGE\bf # 1}\
\noindent \rule[0.05in]{\hsize}{0.02in} \
\vskip -0.1em
}

\begin{multicols}{3}

\bsect{소설}

\book{땡비 (1-4)}
{유안진}
{역사의 음지에서 땡비처럼 살다 스러진
명기(名妓) 고숙향과 의기(義妓) 이조선, 가수 오 여빠와 각설이 사당패들,
그들의 바늘끝 같고 고추 맛같은 사랑과 피눈물로 점철된 질곡의 인생 유전을
그린 대하 소설이다. 시인이자 교수인 저자가 15년간 2000여명의 관련
인물들을 찾아 다니며 집대성한 작품으로 방대 한 스케일, 치밀한 구성으로
특종 한국 문학의 새로운 지평을 제시하고 있다.}
{11.76}
```



\book{남자의 침묵 (1-3)}

{이원호}

{ "밤의 대통령", "황제의 꿈" 등으로 한국

기업 소설계의 최고 작가 이원호의 새로운 장편 소설. 강자와 약자, 대기업과

중소 기업, 고용인과 피고용인 사이의 갈등과 배신, 그 속에서의 경쟁을

그린것으로

야망이 성취되는 순간 명예와 사랑도 함께 한다는 주인공 김영남,

그러나 그 피끓는 야망 반대편에 남겨진 인간 본연의 고독과 번민을 재미있게

그리고 있다. }

{14.72}

\book{길 모퉁이에서 만난 사람}

{양귀자}

{ 원미동을 떠나 서울로 자리 한 후에

작가가 만난 사람들에 관한 소설집. 원숙한 경지로 들 어간 작가의 작품

세계에서 사람이 만들어 놓은 제도나 관습 에 위축당하지 않고 남들과는 다른

방식으로 사는 이들의 스토리를 엮고 있다. }

{11.76}

\book{현장 비평가가 뽑은 올해의 좋은 소설}

{공지영 외}

{1992년 9월 부터 1993년 8월

까지 28종의 문예지를 통해서 발표된 550여편의 중· 단편 소설 중에서 현장

비평가 들이 엄선한 13편의 단편과 2편의 중편 소설 모음집. 공 지영,

구흥서, 김소진, 박상우, 서정인, 신경숙, 윤후명, 이선 외.. }

{14.12}

\book{외로운 여자 (1-5)}

{임선영}

{ 일제 시대 개성 부근에서 자라난 여인 옥수,

그녀를 움아 매던 관습의 굴레가 끊어지면서 강요된 사랑에 빠지는 주인공과

식민지 조선 여인을 사랑한 일본인 쓰오, 옥수의 배신으로 상처받는 우빈,

개성에서 만주에 이어지는 격정적인 사랑의 대하 드라마. 권당

{11.76}

\book{신화}

{한승원}

{ 달의 희유, 구름의 벽, 신화등 3편의 중편 모음집. }

{11.76}

\book{소설 김일성 (1-3)}

{이항구}

{ 북한에서 김일성을 수행 하던 기자 출신의

저자가 김일성이 한 인간으로서 어떤 성격과 가치 의식, 세계관을 지니고

있으며 어떻게 북한의 인민 대중을 이끌며 조국통일 문제를 생각하고

있는지를 밝히는 소설로 김일성 부자, 지도층과 여러 계급의 생활 모습을

수필화한 작품. }

{11.76}

\book{형자와 그 옆사람}

{김채원}

{ 작가의 최초의 장편 소설로 중년에 접어든

여자의 시각에서 본 현대인의 실존과 관계를 회화적으로 그려낸 작품이다. }

{11.76}

\book{나, 고은 (1-3)}

{ 고은}

{시인 고은의 자전 소설로 식민지 시대 한 농부의  
아들로 태어나 식민지 시대, 극한적인 민족 대립의 시대를 거쳐 현재에  
이르기 까지 자기의 전기 내지 회상을 소설 형태로 쓴 글이다.}  
{12.94}

\book{사랑의 절벽}

{경요}

{자유중국 여류작가인 경요 의 베스트 셀러 러브  
로망이다. 1910년대 말로부터 1920년대초에 이르는 중국을 무대로 전통  
사회의 통념과 맞서서 사랑을 쟁취하는 청춘 남녀의 이야기를 구구절절이  
재미있게 엮은 소설 이다 }  
{11.76}

\book{사랑은 파도를 넘어}

{(Vanished) 다니엘 스틸}

{로맨스와 음모를 탁월히  
결합하는 여류작가 다니엘 스틸의 신작 장편소설. 비극적인 사고로 죽은  
아이의 기일 (忌日 )에 주인공 메리얼은 전 남편 찰스와 재회하게 된다. 며칠  
후 재혼 후에 얻은 아들 테디가 실종되면서 사건은 점점 복잡해지게  
되는데... 갈등과 파탄, 만남과 사랑을 재미있게 그리고 있다. }  
{12.94}

\book{연인들의 방(Damage)}

{조세핀 하트 프랑스}

{영화감독 루이 말에 의해

영화화되기도 했던 소설로 완벽했던 한 가정에 어느날 초대 된 위험한 사랑을  
그리고 있다. }  
{12.94}

\book{토탈 호러}

{로버트 블록 외}

{“미래와 외계의 공포” 를 주제로 엮은 단편  
소설 걸작선 으로 일반 독자들이 흔히 생각 하는 공포와는 다른 아주 획기적  
이고 자유 분방한 상상력을 담은 이야기를 수록하고 있다. }  
{11.76}

\book{공포 특급 }

{}

{93편의 현대판 무서운 이야 기 엮음집으로 소재도 참신하고  
짜릿한 반전으로 독자들의 흥미 를 끈다. }  
{9.41}

\book{의태자궁 }

{가토 신스께 외}

{기괴 요염 미스터리. 인류 발전을 위해서라는  
미명하에 벌어지는 인간과 침팬지의 교배, 그로부터 발생하는 새로운 종의  
탄생이 엮는 기괴한 스토리. }  
{12.94}

\book{1994년 신춘 문예 당선 작품집}

{김지연 외}

{각 일간지에 뽑힌 1994년

신춘 문예 당선자들의 시, 시평, 소설, 소설 평과 그들의 당선 소감,  
심사평을 실고 있다. }  
{16.47}

\book{한국 대표 단편 소설선 (상,하)}

{}

{김동인에서

황석영에 이르는 한국 현대 문학 대표 단편 소설 선집. 권당 }

{ 13.65}

\book{술꾼의 아내}

{오정희}

{그만그만하는 사람들이 모여 사는 동네, 일견

평범하고 단조로운 우리와 이웃의 생활 이지만 그 속에 싹틔듯처럼 얽혀 있는

기쁨, 열망, 사랑, 분노의 이야기를 그린 짧은 소설 모음집 }

{12.94}

\bsect{비소설}

\book{대륙의 딸\ (Wild Swans - three daughters of China) (1,2)}

{장영}

{필립의 "대지" 에 비금 간다는 찬사를 받고있는 작품으로 영국 논픽션 부문

NCR상, 작가 협회상을 수상한 베스트 셀러 작품이다. 청조 말부터 일본의

만주 점령, 모택동의 공산 혁명과 문화 대혁명에 이르는 격동의 세월을

살아온 저자의 외할머니, 어머니, 저자 자신에 이르는 파란 만장한 중국

여인들의 슬프고도 아름다운 생생한 삶의 기록이다. 권당 }

{11.76}

\book{나의 아버지 등소평 (1,2)}

{등용}

{중국의 최고 지도자 등소평 의 셋째딸인 저자가 쓴

등소평의 일대기이다. 어린시절 프랑스로 유학을 떠나 고학을 하는 과정

에서부터 위대한 혁명가로 변모 해간 전후 사정을 상세히 묘사 하고 나아가

등소평과 중국 공산당 주요 지도자들인 모택동 과 주은래 및 현대 중국

사상의 주요인물과의 관계도 상세히 설명하고 있다. 그의 가정, 혼인, 성격,

취미들에 대해서도 충분히 묘사하고 있다. 권당 }

{15.29}

\end{multicols}

\end{document}



# A 부록

## 수학기호

도표 A.1: Greek 문자 | 소문자

,	\alpha	4	\iota	%	\varrho
-	\beta	5	\kappa	<	\sigma
.	\gamma	6	\lambda	&	\varsigma
/	\delta	7	\mu	=	\tau
0	\epsilon	8	\nu	>	\upsilon
"	\varepsilon	9	\xi	?	\phi
1	\zeta	o	o	'	\varphi
2	\eta	:	\pi	@	\chi
3	\theta	\$	\varpi	A	\psi
#	\vartheta	;	\rho	!	\omega

도표 A.2: Greek문자 | 대문자

!	<code>\Gamma</code>	\$	<code>\Lambda</code>	'	<code>\Sigma</code>
"	<code>\Delta</code>	#	<code>\Theta</code>	(	<code>\Upsilon</code>
%	<code>\Xi</code>	&	<code>\Pi</code>	)	<code>\Phi</code>
*	<code>\Psi</code>	@	<code>\chi</code>	+	<code>\Omega</code>

도표 A.3: 액센트

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

도표 A.4: Binary Operation 기호

'	\pm	u	\sqcap	▷	\rhd
(	\mp	t	\sqcup	5l	\unlhd
#	\times	_	\vee	6	\unrhd
%	\div	^	\wedge	)	\oplus
\$	\ast	n	\setminusminus	*	\ominus
?	\star	o	\wr	+	\otimes
/	\circ	&	\diamond	,	\oslash
0	\bullet	4	\bigtriangleright	-	\odot
"	\cdot	5	\bigtriangledown	.	\bigcirc
\	\cap	/	\triangleleft	y	\dagger
[	\cup	.	\triangleright	z	\ddagger
]	\oplus	<	\lhd	q	\amalg

도표 A.5: \Log{Like} 기호

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

도표 A.6: 기타 기호

@	<code>\aleph</code>	$\nabla$	<code>\nabla</code>	]	<code>\sharp</code>
$\hbar$	<code>\hbar</code>	$\sqrt{\phantom{x}}$	<code>\surd</code>	$\backslash$	<code>\backslashash</code>
{	<code>\imat</code>	>	<code>\top</code>	@	<code>\partial</code>
	<code>\jmat</code>	?	<code>\bot</code>	1	<code>\infty</code>
`	<code>\ell</code>	k	<code>\kern</code>	2	<code>\Box</code>
}	<code>\wp</code>	$\angle$	<code>\angle</code>	&	<code>\diamond</code>
<	<code>\Re</code>	$\forall$	<code>\forall</code>	4	<code>\triangle</code>
=	<code>\Im</code>	$\exists$	<code>\exists</code>		<code>\clubsuit</code>
0	<code>\rho</code>	:	<code>\neg</code>	}	<code>\diamondsuit</code>
0	<code>\prime</code>	[	<code>\flat</code>	~	<code>\heartsuit</code>
;	<code>\emptyset</code>	\	<code>\natural</code>	P	<code>\spadesuit</code>



도표 A.7: Summation, Intigral etc.

P	\sum	T	\bigcap	J	\bigodot
Q	\prod	S	\bigcup	N	\bigotimes
,	\coprod	F	\bigsqcup	L	\bigoplus
R	\int	W	\bigvee	U	\biguplus
H	\oint	V	\bigwedge		

도표 A.8: 괄호종류

(	)	[	]
f	g	b	c
d	e	h	i
j	k		

도표 A.9: 화살표

A	<code>\leftarrow</code>	!	<code>\rightarrow</code>
(	<code>\Leftarrow</code>	)	<code>\Rrightarrow</code>
A!	<code>\longleftarrow</code>	!!	<code>\longrightarrow</code>
(=	<code>\Longleftarrow</code>	=)	<code>\Longrightarrow</code>
A-	<code>\hookleftarrow</code>	!	<code>\hookrightarrow</code>
(	<code>\leftarrowup</code>	*	<code>\rightarrowup</code>
)	<code>\leftarrowdown</code>	+	<code>\rightarrowdown</code>
,	<code>\leftrightarrow</code>	\$	<code>\leftrightarrow</code>
( )	<code>\longleftrightarrow</code>	A!	<code>\longleftrightarrow</code>
∇	<code>\nabsto</code>	∇!	<code>\longnabsto</code>
↔	<code>\rightleftarrows</code>	;	<code>\leadsto</code>
"	<code>\uparrow</code>	#	<code>\downarrow</code>
*	<code>\Uparrow</code>	+	<code>\Downarrow</code>
	<code>\updownarrow</code>	m	<code>\Updownarrow</code>
%	<code>\nearrow</code>	&	<code>\searrow</code>
-	<code>\nwarrow</code>	.	<code>\swarrow</code>

## B 부록

# 참고 문헌 설명

TEX의 기본 원리와 특별히 plain TEX를 알기 위해서는 Donald Knuth 저서인 [4]을 참조하시요. [1]도 TEX를 빨리 이해 하기위한 좋은 참고 서적이다.

LaTEX는 LaTeX의 저자가 직접쓴 [5]가 필수 지침서이고, LaTeX2e의 의 설명은 [5]의 개정판인 [6] 잘 설명되어 있다. 그외 예도 [8]에 유용한 Tip이 많이 나와 있다.

LaTEX에서 PostScript 폰트 및 그림을 쓰기 위한 필독서로는 [3]가 있고, TEX의 여러 Platform에서의 구성과 유용한 툴들은 [9]를 보시요. [7, 2]등도 Index와 bibliography를 만들기위한 좋은 지침서이다.



# 색인

- [1] Paul W. Abrahams, Karl Berry, and Kathryn A. Hargreaves. `TEX` for the Impatient. Addison Wesley, 1990.
- [2] David J. Buerger. `lATEX` for Scientist and Engineers. McGraw-Hill, 1990.
- [3] Michel Goossens, Frank Mittelbach, and Alexander Samarin. The `lATEX` Companion. Addison-Wesley Pub. Co., 1994.
- [4] Donald E. Knuth. The `TEX`book. Addison-Wesley, 1984.
- [5] Leslie Lamport. `lATEX`: A Document Preparation System. Addison-Wesley, 1986.
- [6] Leslie Lamport. `lATEX`: A Document Preparation System: User's Guide andReference Manual. Addison-Wesley, 1994.
- [7] Oren Patashnik. `BibTEX`ing. Documentation for general `BibTEX` users, January 1988.
- [8] J. Kenneth Shultis. `lATEX` Notes: Practical Tips for preparing Technical Document. Prentice-Hall, 1994.
- [9] Norman Walsh. Making `TEX` work. O'Reilly & Association, Inc, 1994.

# 색인

- \$, 85
- %, 122
- 주프로그램
  - 계속 찾기, 30
  - 글자모양, 31
  - 글자크기, 31
  - 기타선택 설정, 27
  - 끝, 28
  - 다시, 29
  - 대문자로, 30
  - 모두 선택, 29
  - 모두닫기, 35
  - 목차, 35
  - 문단 정렬, 31
  - 문서구조, 32
  - 문자표, 32
  - 바꾸기, 30
  - 반복, 29
  - 복사, 29
  - 붙이기, 29
  - 새화일, 26
  - 색인, 32
  - 소문자로, 30
  - 수식입력(MathType), 33
  - 연속정렬, 35
  - 열기, 26
  - 원레이텍 기본설정, 28
  - 인쇄, 27
  - 인쇄(모두), 27
  - 인쇄종이 형식 설정, 27
  - 자르기, 29
  - 작업중인 창들, 35
  - 저장, 26
  - 저장 끝, 29
  - 저장 닫기, 27
  - 저장(다른이름으로), 27
  - 저장(모두), 26
  - 저장하고 모두닫기, 35
  - 줄로, 31
  - 지우기, 29
  - 찾기, 30
  - 최근 바꾼곳으로, 31

- 취소, 29
- 클립메타(Clipmeta), 33
- 타일정일, 35
- 폰트 설정, 28
- 표시, 31
- 표시한 곳으로, 31
- 항목 정렬, 32
- 화일 읽어넣기, 34
- 화일이름1-9, 29
- BIBTEX, 35
- DVI 보기/ 인쇄, 34
- HTML 입력, 27
- HTML 출력, 27
- LaTeX 기호입력, 32
- LaTeX 명령입력, 33
- LOG 화일 보기, 34
- MAKEINDEX, 35
- TeX 컴파일, 34
- WinTex는?, 36
- 각주
  - minipage에서, 149
- 경고, 162
- 단면출력, 123
- 머릿말, 122
- 목차만들기, 121
- 문서 종류, 123
- 배치
  - 텍스트, 125
  - minipage, 147
  - 부록, 126
  - 새로운 page로, 121
  - 색인만들기, 123
  - 양면출력, 124
  - 에러, 161
  - 어백, 125
  - 요약, 120
  - 이단출력, 145
  - 장, 119
  - 저서목록, 131
    - 만들기, 136, 137
  - 절, 119
  - 제목, 120
  - 참고문헌, 131
    - 만들기, 136, 137
  - 초록, 120
  - 큰 문서 짜기, 126
  - 텍스트의 크기, 125
  - 페이지번호, 121
  - 가운데로 정렬, 110
  - 검색메뉴, 30
  - 고딕체, 79
  - 괄호, 91
  - 궁서체, 79
  - 그래픽스 필터, 51
  - 그림 화일 넣기, 50
  - 그림이 있는 문서, 23

글자꼴, 79  
 글자크기, 80  
 기호쌓기, 92  
 도움말 메뉴, 35  
 디나루체, 80  
 명조체, 79  
 밑줄, 93  
 배열, 90  
 배치  
     표, 106  
     tabbing, 103  
 변환기: hlx2tex.exe, 14  
 분수의 표현, 87  
 수식, 85  
     번호, 89  
 수식이 있는 문서, 22  
 아랫첨자, 86  
 영문필기체, 93  
 오른쪽끝 맞춤, 111  
 왼쪽끝 맞춤, 110  
 윗첨자, 86  
 이탤릭체, 79  
 인용문, 114  
 입력 화일, 14  
 입력메뉴, 31  
 조판메뉴, 34  
 창메뉴, 35  
 편집메뉴, 29  
 폰트의 추가, 65  
 표, 104  
 표제, 106  
 프로그램의 시작, 19  
 화일메뉴, 26  
 확장자:extension, 14  
  
 abstract, 120  
 ACM proceeding style option, 145  
 appendix, 126  
 array, 90  
     delimiter, 91  
  
 bibliography, 131  
     making of, 136  
     making with BibTeX, 136  
     making without BibTeX, 137  
     thebibliography, 137  
 BibTeX, 131  
 boldface type, 79  
  
 cal, 93  
 caligraphic letter, 93  
 caption, 106  
 center, 110  
 chapter, 119  
 cite, 135  
 clearpage, 130  
 dine, 104  
 dipmeta, 51



- counter, 158
  - name, 158
- database
  - bibliography, 131
- delimiter, 91
- description
  - customized, 113
- description, 112
- displaymath, 86
- dithering, 54
- document assembly, 126
- documentstyle option, 123
- DVIWIN
  - 경고음, 59
  - 끝, 56
  - 다른 페이지, 56
  - 로그 화일 보기, 59
  - 색깔 조정, 61
  - 선택 메뉴, 57
  - 수직 유지, 59
  - 수평 유지, 59
  - 자동환경 저장, 59
  - 페이지 메뉴, 56
  - 페이지 크기, 61
  - 폰트 메모리, 60
  - 폰트 추적, 58
  - 폰트찾기, 61
  - 표시자, 58
  - 프린트, 56, 57
  - 해상도 메뉴, 57
  - 해상도지정, 60
  - 화면 움직이기 지정, 60
  - 화일 메뉴, 56
  - 확대 메뉴, 57
  - 확대경, 61
  - 흑백, 59
  - 끝(선택사항 저장), 56
  - PK 폰트쓰기, 59
  - So, TeX International, 61
- emphaasized type, 79
- entry
  - bibliography, 132
- enumerate, 111
- environment
  - center, 110
  - description, 112
  - enumerate, 111
  - gure, 106
  - .ushleft, 110
  - .ushright, 111
  - minipage, 146
  - quotation, 114
  - quote, 114
  - tabbing, 101
  - table, 104, 106
  - verbatim, 117

- verse, 115
- eqnarray, 88
- equation, 86
- error, 161
- extension, 14
- eld
  - bibliography, 134
- . eqn, 86
- . ushleft, 110
- . ushright, 111
- font, 79
  - loading of, 81
- footnote
  - minipage, 149
- frac, 87
- glossary, 139
- gothic type, 79
- half-toning, 54
- header, 124
- hline, 92, 104
- IEEE proceeding style option, 145
- include, 130
- index, 140
- input, 130
- input - le, 14
- italic type, 79
- key
  - bibliography, 132
- kill, 103
- layout, 125
- ldots, 88
- list of - gures, 121
- list of tables, 121
- loading font, 81
- makeindex, 123, 143
- maketitle, 120
- margin, 125
- math, 85
- minipage, 146
- multicolumn, 91, 104
- myoungjo type, 79
- newcommand, 155
- newenvironment, 156
- newpage, 121, 130
- newtheorem, 157
- noindent, 126
- number
  - page, 121
- numbering
  - in section, 157
  - of equation, 89
- one-sided output, 123

- option
  - documentstyle, 123
  - . eqn, see . eqn
  - makeindex, 123
  - pagestyle, 124
  - titlepage, 120, 123
  - twocolumn, 123
  - twoside, 123
- overbrace, 93
- overline, 93
- page, 124
  - counter, 158
  - heading, 124
  - layout, 125
  - set counter, 158
- pagenumbering, 121
- pagestyle, 124
- positioning
  - in tabbing, 103
  - of minipage, 147
  - of table, 106
- preamble, 122
- preface, 122
- proceeding, 145
- quotation, 114
- quote, 114
- reference, 135
- renewcommand, 156
- romam type, 79
- san serif type, 79
- section, 119
- set counter, 158
- size
  - font size, 80
  - of text, 125
- slanted type, 79
- small caps type, 79
- stacking symbol, 92
- subscript, 86
- subsection, 119
- superscript, 86
- tabbing, 101
- tabbing command, 103
- table, 104, 106
- table of contents, 121
- tabular, 104
- thebibliography, 137
- title, 120
- titlepage, 120, 123
- two-sided output, 123
- twocolumn, 123, 145
  - proceeding, 145
- twoside, 123
- typewriter type, 79

underbrace, 93

underline, 93

verb, 117

verbatim, 117

verse, 115

warning, 162

widehat, 93

widetilde, 93