

# 논문 작성을 위한 기초 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>

이남형, 조남운, 한원용, 민병길, 김형중

고려대학교 경제학과

2012. 7. 27

Ver. 1.6

## 요 약

이 안내서는 L<sup>A</sup>T<sub>E</sub>X을 사용하여 학위 논문을 쓰려는 학생에게 기초적인 도움을 주기 위한 것이다. 이 글은 워드 프로세서를 사용하다 L<sup>A</sup>T<sub>E</sub>X으로 전환하려는 사람을 대상으로 하므로, 어느 정도의 수준에 도달한 사용자에게는 부적합하다. 또 한글 논문 작성을 기준으로 작성되었다. 또한 저자들의 경험을 바탕으로 한 것이므로 사용자는 이 안내서에서 제시한 해법이외에도 자기 나름의 해법을 찾아갈수 있다. 이남형은 1, 2, 3, 6, 7장을, 조남운은 4, 5장과 8장의 초안을, 한원용은 9장을, 민병길을 8장을, 김형종은 LyX와 관련된 내용을 작성했다. 안내서의 오류는 <latex.ko@gmail.com>으로 연락주시기 바란다.

T<sub>E</sub>X을 만든 Donald Knuth, L<sup>A</sup>T<sub>E</sub>X을 개발한 Leslie Lamport, 그리고 한글화 구현 및 사용자 편의성을 높이기 위해 여러모로 애쓰시고, 활발한 질의 응답으로 문제 해결을 주시는 ktug의 많은 회원들께 감사드린다.

# 목 차

변경사항	iv
<b>제 1 장   <math>\text{\LaTeX}</math> 소개</b>	<b>1</b>
1.1 문서의 구조와 $\text{\LaTeX}$ 의 장점 . . . . .	1
1.1.1 문서의 구조 . . . . .	1
1.1.2 $\text{\LaTeX}$ 의 특징 . . . . .	1
1.2 $\text{\LaTeX}$ 설치 . . . . .	3
1.2.1 LyX의 설치 및 한글설정 . . . . .	4
1.2.1.1 Mac에서 LyX 설치 . . . . .	5
1.2.1.2 Windows에서 LyX 설치 . . . . .	7
1.2.1.3 LyX에서 한글설정 . . . . .	7
1.2.1.4 LyX에서 $\text{\TeX}$ 으로 . . . . .	7
1.3 $\text{\LaTeX}$ 시스템 . . . . .	9
1.3.1 $\text{\LaTeX}$ 시스템에서 사용하는 파일 . . . . .	9
1.3.2 $\text{\LaTeX}$ 시스템에서 생성되는 파일 . . . . .	9
1.4 $\text{\LaTeX}$ 관련 문서 및 홈페이지 . . . . .	10
1.4.1 관련 문서 . . . . .	10
1.4.2 관련 홈페이지 . . . . .	11
<b>제 2 장   <math>\text{\LaTeX}</math> 문서의 기초</b>	<b>12</b>
2.1 $\text{\LaTeX}$ 명령어의 기초와 컴파일 . . . . .	12
2.2 작업 폴더와 파일 인코딩 . . . . .	13

2.3	프리앰블(pre-amble)	13
2.3.1	문서 클래스(document class)	13
2.3.2	패키지(usepackage)	14
2.4	한글구현	15
2.5	표제 작성	16
2.5.1	제목, 저자, 작성일 작성	16
2.5.2	초록 작성	16
2.5.3	목차 작성	16
2.6	본문 작성	17
2.6.1	장, 절의 설정	17
2.6.2	본문	18
2.6.3	각주와 난외주	19
2.6.4	인용	19
2.6.5	참고문헌, 표, 그림 넣기	19
2.6.6	상호참조	20
2.6.7	프로그램 코드를 그대로 입력하기	20
2.6.8	각 장을 파일별로 나누기	20
2.7	개조식 문서 작성	21
2.7.1	개조식 문서 작성을 위한 명령어	21
2.7.2	모양 바꾸기	23
<b>제 3 장</b>	<b>참고문헌 작성</b>	<b>24</b>
3.1	BibTeX의 개요	24
3.2	참고문헌 데이터베이스 만들기	24
3.3	인용 방법 및 참고문헌 목록 스타일	27
3.3.1	참고문헌 관리 프로그램	30
<b>제 4 장</b>	<b>표와 그림</b>	<b>31</b>
4.1	그림 넣기	31
4.1.1	예제 1: 간단한 그림 넣기	31

4.2	표 만들기 . . . . .	34
4.2.1	간단한 표의 예 . . . . .	34
4.2.2	조금 더 복잡한 표 . . . . .	36
4.2.2.1	열병합 표 . . . . .	36
4.2.2.2	행병합 표 . . . . .	37
4.2.2.3	표 속에 각주 넣기 . . . . .	38
4.2.2.4	Excel2Latex의 사용 . . . . .	40
4.2.2.5	csvsimple 패키지의 사용 . . . . .	40
<b>제 5 장</b>	<b>수식</b>	<b>42</b>
5.1	수식모드 . . . . .	42
5.2	수식의 표현법 . . . . .	43
5.2.1	기초적 용법 . . . . .	43
5.2.2	첨자 있는 화살표 . . . . .	44
5.2.3	적분 . . . . .	44
5.2.4	경우의 수 . . . . .	45
5.2.5	행렬 . . . . .	45
5.2.6	align 환경 . . . . .	46
5.2.7	극한, 확률 . . . . .	46
5.2.8	LyX에서 수식입력 . . . . .	47
<b>제 6 장</b>	<b>L<sup>A</sup>T<sub>E</sub>X 문서의 편집 기초</b>	<b>49</b>
6.1	글자 모양 . . . . .	49
6.1.1	글꼴 바꾸기 . . . . .	49
6.1.2	글꼴 크기 . . . . .	50
6.1.3	글꼴 모양 . . . . .	50
6.1.4	밑줄 긋기 . . . . .	51
6.2	문단 모양 . . . . .	52
6.2.1	들여쓰기 . . . . .	52
6.2.2	줄 간격 . . . . .	52

6.2.3	문단 간격	53
6.2.4	정렬	53
6.3	편집용지	54
6.3.1	문서 클래스의 옵션 활용	54
6.3.2	문서여백	54
6.3.2.1	setlength 명령어의 활용	54
6.3.2.2	geometry 패키지의 활용	55
6.4	머리말/꼬리말	55
6.5	쪽 번호 매기기/새 번호로 시작	56
6.6	줄 나누기/쪽 나누기	57
6.6.1	줄 나누기	57
6.6.2	쪽 나누기	57
6.7	LyX에서의 편집	57
<b>제 7 장</b>	<b>문서 스타일 바꾸기-명령과 클래스 작성</b>	<b>59</b>
7.1	명령어의 정의	59
7.1.1	대상 이름 바꾸기	60
7.1.2	장/절 편집하기	61
7.1.3	수학 환경 지정하기	62
7.1.4	float 패키지의 사용	63
7.2	클래스 파일 만들기	64
<b>제 8 장</b>	<b>프리젠테이션용 파일 만들기: Beamer</b>	<b>66</b>
8.1	beamer 서식	67
8.2	프레임	67
8.2.1	제목 만들기	68
8.2.2	목차 만들기	68
8.3	프레임 구조	68
8.3.1	다단 편집	69
8.3.2	블록 설정	70

8.4	절 편성	70
8.5	글꼴 모양	70
8.5.1	프로그램 코드나 공식 입력	71
8.5.2	글자크기와 글꼴 테마	71
8.5.3	글꼴 묶음	72
8.6	프레임 정렬, 줄간격, 띄워쓰기	72
8.6.1	프레임 상하정렬	72
8.6.2	프레임 좌우 정렬	72
8.6.3	줄간격과 띄워쓰기	73
8.7	개조식	73
8.7.1	itemize	73
8.7.2	Enumerate	74
8.7.3	Description	74
8.8	오버레이	75
8.8.1	오버레이와 명령어	76
8.8.2	오버레이 환경	77
8.9	표와 그림	78
8.9.1	표 만들기	78
8.9.2	표를 오버레이 시키기	78
8.9.3	그림 삽입	79
8.10	테마	79
8.10.1	전체 테마	79
8.10.2	색 테마	80
8.11	유인물 출력	80
8.12	LyX에서 Beamer 구현	80
<b>제 9 장</b>	<b>그래프 그리기</b>	<b>82</b>
9.1	curves 패키지의 활용	82
9.2	PSTricks의 활용	84

9.2.1	설치 및 패키지 사용 지정 . . . . .	84
9.2.2	컴파일링과 주의사항 . . . . .	84
9.2.3	PSTricks 활용의 기초 . . . . .	85
9.2.3.1	그래프 그리기 . . . . .	85
9.2.3.2	에지워스 상자 그리기 . . . . .	87
9.2.3.3	선분 그리기 . . . . .	88
9.2.3.4	그래프 2개 그리기 . . . . .	88
9.2.3.5	함수를 이용한 그래프 . . . . .	89
9.2.3.6	음함수 그리기; psplotImp . . . . .	90
9.2.3.7	무차별곡선과 예산제약 그래프 . . . . .	90
9.3	LaTeXDraw의 활용 . . . . .	91
9.3.1	LaTeXDraw . . . . .	91
9.3.2	설치 . . . . .	91
9.3.3	eps 파일 생성 . . . . .	92
9.4	그외의 사항들 . . . . .	94
<b>제 10 장</b>	<b>경제학을 위한 <math>\text{\LaTeX}</math></b>	<b>95</b>
10.1	게임 보수표 및 게임 나무의 작성 . . . . .	95
10.1.1	전략형 게임 보수표의 작성 . . . . .	95
10.1.2	확장형 게임 나무의 작성 . . . . .	96
10.2	Stata 결과값 불러오기 . . . . .	98
10.2.1	sutex, outtex 명령어 . . . . .	98
10.2.2	Stata 응용 및 기타 통계 패키지 . . . . .	99
10.3	웹의 정보를 활용하여 참고문헌 달기 . . . . .	100
10.4	논문용 패키지 활용하기 . . . . .	100



# 그림 목차

1.1	LyX의 사용자환경 . . . . .	5
1.2	LyX 인코딩 설정화면 . . . . .	6
1.3	LyX 한글 구현 설정화면 . . . . .	8
4.1	TEX이 뭐야? 먹는거야? . . . . .	32
4.2	상대크기를 폭의 50%로 축소하면서 반시계방향으로 45도 꺾음 . . . . .	34
4.3	Excel2Latex . . . . .	40
5.1	LyX에서 수식입력 . . . . .	48
9.1	curves.sty를 사용한 예산제약식과 무차별 곡선 . . . . .	83
10.1	확장형 게임 . . . . .	98

# 표 목차

2.1	장절 수준 및 번호 . . . . .	17
3.1	표준 BibTeX 참고문헌 종류 입력값 . . . . .	25
3.2	표준 BibTeX 참고문헌 종류별 필수 필드 . . . . .	26
3.3	기본 참고문헌 스타일 . . . . .	28
3.4	인용 명령어 용례 . . . . .	29
4.1	열병합 표 . . . . .	36
4.2	행병합 표 . . . . .	37
6.1	ko.TeX 기본 글꼴 목록 . . . . .	49
6.2	ko.TeX 문서 한글 기본 설정 . . . . .	50
6.3	글자 모양 . . . . .	50
6.4	글자 크기 . . . . .	51
6.5	용지의 각 부분별 이름 . . . . .	56
7.1	float 패키지의 스타일과 성격 . . . . .	64
8.1	beamer의 글꼴 테마 . . . . .	71
8.2	기본 글꼴 묶음 종류 . . . . .	72
8.3	오버레이 사용의 기본 예 . . . . .	76
8.4	오버레이의 응용 . . . . .	77
10.1	전략형 게임 보수표 . . . . .	96

10.2 Stata 결과의 L <sup>A</sup> T <sub>E</sub> X 전환 명령어 . . . . .	99
---	----

# 변경사항

2012년 7월 25일

## 추가

3쪽 TnXTeX 설명 추가.

5쪽 LyX의 장단점 설명 추가.

5쪽 LyX에서 한글 설치 설명 추가.

7쪽 Mac에서 LyX 설정 방법 설명 추가.

40쪽 csvsimple 패키지 사용법 추가.

## 수정

3쪽 T<sub>E</sub>XLive 2011로 변경.

7쪽 LyX의 한글 환경 관련 내용 수정.

15쪽 윈도우즈 상의 스타일 파일 저장 경로 수정.

33쪽 ‘다만, EPS확장자는 지원하지 않으므로 미리 pdf나 jpg 등으로 변환시켜두어야 한다.’ 삭제

84쪽 PSTricks 설치 방법 수정.

91쪽 LaTeXDraw 중심으로 재서술.

## 2010년 8월 12일

### 추가

66쪽 beamer 문서 소개 추가.

75쪽 예제 변경 및 설명 추가.

76쪽 예제 변경 및 설명 추가.

77쪽 예제 및 설명 추가.

### 수정

76쪽 예제 오타자 수정.

## 2010년 8월 10일

### 추가

1쪽 각주에  $\text{\TeX}$ 의 역사 및 특징을 소개하는 url 추가.

4쪽  $\text{\LyX}$  설치 추가.

11쪽 Andy Roberts의 홈페이지 추가.

29쪽  $\text{\LyX}$ 에서의 참고문헌 형식 관리 추가.

30쪽 참고문헌 관리 프로그램 소개.

57쪽  $\text{\LyX}$ 에서 문서 편집 요령 추가.

80쪽  $\text{\LyX}$ 에서의 Beamer 사용법 추가.

91쪽  $\text{\LaTeXDraw}$  소개 추가.

95쪽 경제학을 위한  $\text{\LaTeX}$  추가: `sgamevar` 등을 이용한 게임 보수표 및 트리 작성 이동, Stata 등의 결과값을  $\text{\LaTeX}$ 용으로 전환하기 추가 등.

### 수정

제목 ‘학위논문 작성’ 에서 ‘논문 작성’으로 수정.

8장 내용 수정.

1쪽 장 제목 수정.

11쪽 L<sup>A</sup>T<sub>E</sub>X for Economics의 url 수정.

15쪽 Windows에서의 패키지 추가 방법 수정. KC 2007에서의 설치는 각주 처리.

15쪽 Windows에서의 패키지 및 스타일 수동 설치 경로 수정.

16쪽 Texworks 컴파일로 수정.

24쪽 Windows에서의 .bib 파일 절대 경로 수정. KC 2007에서의 설치는 각주 처리.

28쪽 Windows에서의 .bst 파일 절대 경로 수정. KC 2007에서의 설치는 각주 처리.

30쪽 Texworks 컴파일로 수정.

61쪽 내용 수정.

61쪽 내용 수정.

## 2010년 8월 6일

### 수정

31쪽 각주 수정.

31쪽 url 수정, bb에 대한 설명 삭제, 그림파일 abcd.jpg에서 cat.jpg로 교체.

32쪽 사소한 교정.

42쪽 사소한 교정.

### 추가

31쪽 width, height에 대한 설명 추가.

34쪽 사소한 교정들.

## 2010년 7월 27일

### 수정

3쪽 설치 url, ko.T<sub>E</sub>X 설치, 추천 에디터 수정.

13쪽 Windows의 컴파일 방식을 KCMenu에서 Texworks를 사용하는 것으로 서술 변경.

13쪽 Emeditor에서의 인코딩 설정 방법을 Texworks의 인코딩 설정으로 서술 변경.

85쪽 Texshop에서의 PSTricks 컴파일링 방법 수정.

## 2008년 9월 22일

### 추가

29쪽 Osborne 교수가 만든 te 스타일 추가 소개.

## 2008년 2월 29일

### 추가

9쪽 L<sup>A</sup>T<sub>E</sub>X 시스템 소개하는 1.3절 추가.

13쪽 KCMenu와 TeXShop의 컴파일 방법 추가.

14쪽 패키지 설치 관련 내용 추가.

18쪽 이렇게 하면 목차에 장이 나타나지 않을 것이다. 이 경우 다음의 명령어를 사용하면 된다.

```
\setcounter{chapter}{장번호 재설정}
```

```
\chapter*{장 제목}
```

```
\addcontentsline{toc}{chapter}{*가 붙은 장 제목}
```

30쪽 KCMenu와 TeXShop의 BiB<sub>T</sub>E<sub>X</sub> 컴파일 방법 추가.

31쪽 graphicx 패키지의 사용을 지정한다.

32쪽 그림이 제대로 들어가지 않고 에러가 난다면, 그림의 크기(bb: bounding box)를 잡아주어야 한다. bb=0 0 600 600은 그림의 크기를 지정하는 것인데, 첫번째와 세번째는 가로 크기, 두번째와 네번째는 세로 크기를 지정한다. 예를 들어 그림의 크기가 1024X768 이라면, bb=0 0 1024 768이 된다.

38쪽 표 속에 각주 넣기 4.2.2.3절 추가.

40쪽 excel2latex을 사용한 표 만들기 4.2.2.4절 추가.

82쪽 curves.sty 패키지를 활용한 그래프 작성에 관한 9.1절 추가.

95쪽 sgamevar.sty를 다운 받으면 이 파일을 열고, 가장 마지막 두 줄을 삭제한다. 이 부분은 인코딩 충돌을 일으킬 것이다.

수정

3쪽 설치 url 수정

63쪽

```
\newenvironment{proof}[1][증명]{\noindent\textbf{#1.} }\{\rule{0.5em}{0.5em}}
```

65쪽

```
\NeedsTeXFormat{LaTeX2e}
```

84쪽 Windows의 경우, 다운받은 압축파일을 풀고, dvips 폴더에 있는 파일은  
\\KC2007\\TeX\\Share\\texmf-dist\\dvips\\pstricks 폴더에 옮겨 놓는다. (2012년 7월  
삭제)

96쪽 단, 다음의 순서를 지켜야 한다. (2012년 7월 삭제)

```
\usepackage{pstcol}
```

```
\usepackage{pstricks}
```



# 제 1 장 L<sup>A</sup>T<sub>E</sub>X 소개

## 1.1 문서의 구조와 L<sup>A</sup>T<sub>E</sub>X의 장점

### 1.1.1 문서의 구조

- 표제(title): 누가(저자), 언제(작성일), 무엇을(제목) 썼는가를 보여주는 부분
- 목차(contents): 페이지를 지정하는 것과 더불어, 제목을 통해 글의 내용을, 그리고 순서를 통해 글의 구성을 한눈에 알 수 있게 한다.
- 제목(headline): 해당 부분에서 무엇을 말하고 있는지 명확하게 보여준다.
- 단락(paragraph): 하나의 화제를 다루고 있는 하나의 부분
- 들여쓰기(indentation): 관습적으로 단락이 시작하는 부분을 들여쓰므로써, 단락의 구분을 분명히 해준다. 서양문서에서는 가장 첫 단락에 한해 들여쓰기를 하지 않기도 한다.
- 구두점(punctuation): 문장의 종결을 표시하며, 어떤 구두점을 사용하느냐에 따라 문장의 의미를 결정하기도 한다.
- 주석(note): 이해를 돕기 위해 보충 정보를 제공하는 부분이다. 각주, 미주, 난외주 등이 이에 포함된다.

### 1.1.2 L<sup>A</sup>T<sub>E</sub>X의 특징<sup>1</sup>

- 글쓰기에 집중할 수 있게 해준다.

---

<sup>1</sup>이는 전적으로 이남형의 견해이다. T<sub>E</sub>X의 역사 및 특징은 다음 홈페이지를 참고하라([http://www.ctan.org/what\\_is\\_tex.html](http://www.ctan.org/what_is_tex.html))

- 논문은 형식이 정해진 논리적인 글이다. 형식이 정해져 있다는 것은, 앞에서 말한 문서의 구조가 학문공동체의 합의된 일반 원칙을 따른다는 것을 의미한다.
  - 워드 프로세서는 논문보다는 일반적인 글쓰기에 적합한 애플리케이션이다.
  - T<sub>E</sub>X의 목표는 저자는 내용을 쓰고, 조판기인 T<sub>E</sub>X이 저자의 명령(command)에 따라 문서의 형식을 처리하는 것이다.
  - L<sup>A</sup>T<sub>E</sub>X은 T<sub>E</sub>X의 복잡한 명령을 좀 더 간단하게 처리할 수 있도록 한다.
  - L<sup>A</sup>T<sub>E</sub>X은 정해진 형식에 따라 내용을 보여주는 데 적합하다.
  - 그러므로 저자는 글에 집중하고, 부차적인 편집은 덜 신경써도 된다.
  - 그리고 L<sup>A</sup>T<sub>E</sub>X의 결과물이 전문가 수준으로 디자인된 것처럼 나온다.
- 논리적인 글쓰기에 적합하다.
    - L<sup>A</sup>T<sub>E</sub>X은 글의 구성을 먼저 생각하도록 강제한다.
    - L<sup>A</sup>T<sub>E</sub>X은 장, 절, 소절 등의 구조를 지정하지 않고, 메모를 하듯 글을 쓰면 결과물이 보기 좋게 나오지 않는다.
    - L<sup>A</sup>T<sub>E</sub>X으로 비체계적이고 구조화되지 않은 문서를 작성하기는 어렵다.
    - 논문은 장, 절, 문단이 체계적이고 논리적으로 배치된 글이다.
  - L<sup>A</sup>T<sub>E</sub>X은 쉽다.
    - 아래아 한글이나 MS Word 등 보는 대로 쓸 수 있는 워드 프로세서<sup>2</sup>에 익숙한 사람이 L<sup>A</sup>T<sub>E</sub>X을 처음 보면 어렵게 느껴진다.
    - ‘원고의 편집 → 조판(typeset) → 미리보기(preview) → 출력/인쇄(print)’의 순서로 글이 만들어지는 데, 일반 워드프로세서는 위 작업을 한 번에 처리하지만, L<sup>A</sup>T<sub>E</sub>X은 단계별로 처리하고, 각 단계별로 상이한 애플리케이션을 사용하므로 어렵게 보인다.
    - 또 명령어로 가득찬 문서를 처음부터 쉽게 읽기는 어렵다.

---

<sup>2</sup>WYSIWYG: What you see is what you get. 최종 출력물의 모양을 화면으로 보면서 작업할 수 있다.

- 그렇지만 워드 프로세서에 있는 각주, 스타일, 상호참조, 개요, 목차, 참고문헌 등의 고급 기능을 사용하는 것은  $\text{\LaTeX}$ 이 더 쉽다.
- 그리고 이 것들은 논문에 필수적인 요소들이다.

## 1.2 $\text{\LaTeX}$ 설치

- $\text{\LaTeX}$  설치하는 다음 주소를 참조하라.

<http://www.ktug.or.kr/xe/?mid=install>

- Windows의 경우, TeXLive 2011<sup>3</sup>을 다운로드 받아 실행시키면 자동으로 설치가 진행된다.
- Mac의 경우, MacTeX 2011을 다운 받아 설치한 후, 한글 패키지인 kotex를 설치하면 된다.
- $\text{\LaTeX}$ 은 조판을 위한 컴퓨터 언어라고 생각하면 된다. 따라서 이에 특화된 별도의 편집기가 필요하고, 결과물을 볼 수 있는 뷰어가 있어야 한다.
- 이는 인터넷을 사용할 때, html 언어가 있고, html 전문 편집기와 결과물을 볼 수 있는 웹브라우저가 필요한 것과 마찬가지로이다.
- 그리고 여기에 한글을 구현하기 위한 별도의 파일이 추가되어야 한다. 여기에는 TeX live 2011과 한글 구현용 ko.TeX, 편집기 겸 조판기 겸 뷰어인 TeXworks 및 관련 파일이 포함되어 있다. 패키지와 별도로 pdf나 ps 뷰어가 필요할 수도 있다. Adobe의 Acrobat이 대표적인 관련 애플리케이션이다.
- KTUG 사이트에서는 이 모든 기능을 한 번에 해결할 수 있는 패키지 ko.TeX Live 2011을 받을 수 있다. 여기에는 ko.TeX, 편집기 겸 조판기 겸 뷰어인 TeXworks가 포함되어 있다.
- 윈도우즈에서는 무설치 버전인 TnXTeX을 사용할 수도 있다. TnXTeX은 사용에 필요한 최소한의 패키지를 포함하고 있어 상대적으로 용량이 작다. 추가적으로 필요한 패키지는 TeXworks의 콘솔을 사용하여 손쉽게 추가할 수 있다.

---

<sup>3</sup>TeXLive 뒤의 숫자는 년도 및 버전을 의미하며, 보통 여름 무렵에 당해년도 버전이 나온다.

- KTUG 사이트에는 또한 Mac용 설치 패키지도 다운받을 수 있다. 여기에는 MacTeX과 한글 구현용 ko.TEX, 편집기 겸 조판기 겸 뷰어인 TeXShop이 포함되어 있다.
- L<sup>A</sup>T<sub>E</sub>X도 WYSIWYG 방식의 편집기가 있으며 LyX(오픈소스)과 Scientific Word(유료)가 대표적이다. LyX에 대해서는 관련된 부분에서 중간 중간 소개할 것이다.
- 현재 Unicode 사용이 가능한 편집기는 다음과 같다. Windows에서는 TeXworks, Mac에서는 TeXshop이 대표적이며, 본 안내서도 이 둘을 중심으로 다룰 것이다.
  - WinEdt (유료, Windows): <http://www.winedt.com/>
  - WinShell (무료, Windows): <http://www.winshell.org/>
  - TeXshop (오픈소스, Mac): <http://www.uoregon.edu/~koch/texshop/>
  - Notepad++ (오픈소스, Windows): <http://www.notepad-plus-plus.org/>
  - TeXMaker (오픈소스, cross-platform): <http://www.xmlmath.net/texmaker/>
  - TeXworks (오픈소스, cross-platform): <http://en.wikipedia.org/wiki/TeXworks>
  - LyX (오픈소스, cross-platform): <http://www.lyx.org/>
  - Scientific Word (유료): <http://www.mackichan.com/>

### 1.2.1 LyX의 설치 및 한글설정

LyX은 직관적인 사용자 인터페이스를 가지고 있어 MS워드나 한글등의 문서편집기 사용자들이 쉽게 T<sub>E</sub>X으로 전환할 수 있도록 도와준다. Scientific Word도 유사한 사용자 인터페이스를 제공하지만 유료이다. Scientific Word가 LyX에 비해 기능적으로 특별하게 우월하다고 생각되는 부분은 찾기 어려우므로 여기서는 오픈소스인 LyX을 중심으로 설명한다.

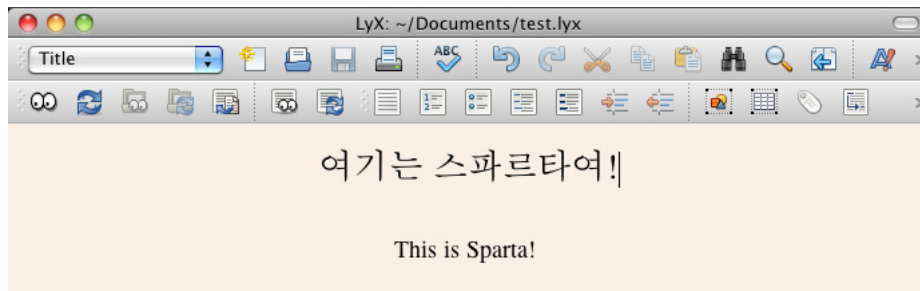


그림 1.1: LyX의 사용자환경

앞서 언급한 것처럼 일반적인 문서편집기와 유사한 사용자 경험을 제공한다는 점 외에도 능숙하게  $\text{\TeX}$ 을 다룰 수 있는 사용자들에게 도움이 될 수 있는 특징들 역시 가지고 있다.<sup>4</sup>

단점은 Scientific Word와 달리 텍파일(.tex)을 직접 수정하지 않고 자체적인 릿파일(.lyx)을 생성하는 점이다. 하지만 릿파일을 텍파일로 변환하면 Scientific Word처럼 일반적이지 않은 문법을 사용하지는 않는다는 점을 알 수 있다.

이하에서 LyX 설치법을 소개한다.<sup>5</sup>

#### 1.2.1.1 Mac에서 LyX 설치

Mac에서 LyX을 설치한다면 자신의 Mac이 인텔맥인지 아닌지를 파악한다. LyX은 타이거(OSX Tiger)와 그 이후의 버전 각각을 위한 dmg가 구분되어 있다. 한글 사용을 위해서는 ‘도은이아빠’님이 제공하시는 파일로 설치하는 것을 권장한다.<sup>6</sup>

Mac $\text{\TeX}$ 과 한글 환경이 설치된 Mac을 가정한다. LyX을 다운로드하고 ‘응용 프로그램’에 넣어둔다. LyX → Preferences → Output → LaTeX에 들어가서 ‘Use LaTeX font

<sup>4</sup>예를 들어 한글로 beamer를 작성하는 경우에 파워포인트로 작업하는 것과 유사한 환경을 제공할 수 있다. 그리고 오픈오피스 계열의 스프레드시트를 사용하면 MS엑셀에서 MS워드로 표를 붙이는 것과 같은 작업들 구현할 수 있다.  $\text{\TeX}$ 에 숙달된 사용자라도 급하게 beamer나 표를 만드는 것은 제법 시간이 걸리는 작업이다. LyX은 이러한 작업들을 깨끗하지는 않지만 빨리 할 수 있도록 도와준다.

<sup>5</sup>설치에 사용할 파일은 ‘도은이아빠’님이 수정하신 파일을 권장한다. lyx.org에서 제공하는 파일을 사용하여도 한글 구현이 가능하지만 여러가지 설정을 스스로 해야한다: 예를 들어 oblivoir 클래스의 설치. 다음의 링크 <http://faq.ktug.or.kr/faq/Karnes/LyX20>를 참조하기 바란다.

<sup>6</sup>동일하게 <http://faq.ktug.or.kr/faq/Karnes/LyX20>를 참조한다.

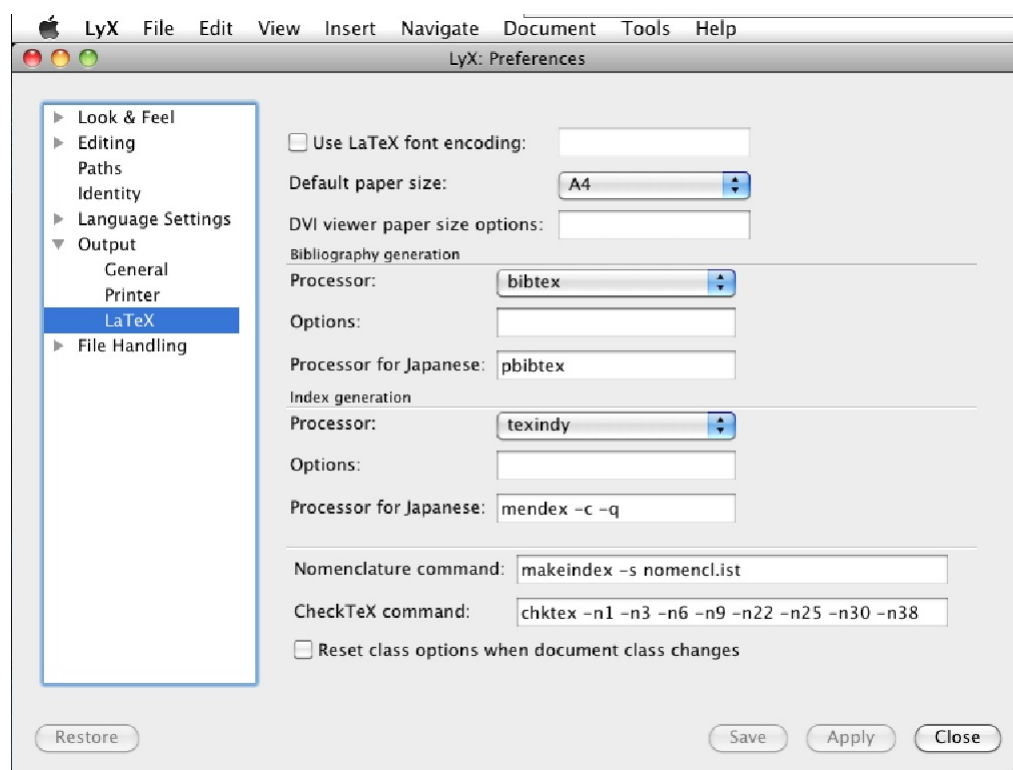


그림 1.2: LyX 인코딩 설정화면

encoding:’에 체크를 해제하고 ‘save’한다.

#### 1.2.1.2 Windows에서 LyX 설치

우선 ko.T<sub>E</sub>X Live 2011이 설치된 윈도우즈 머신을 가정한다. 만약 ko.T<sub>E</sub>X Live 2011이 설치되지 않은 경우라면 LyX 설치과정에서 MiK<sub>T</sub>E<sub>X</sub>의 설치를 물어본다. MiK<sub>T</sub>E<sub>X</sub>을 사용하면 한글 사용을 위한 작업은 스스로 책임져야 한다. 영어만 사용하려는 경우는 MiK<sub>T</sub>E<sub>X</sub>을 설치해도 문제없다.

윈도우즈용 LyX은 설치과정에서 T<sub>E</sub>X의 경로를 물어본다. 사용자가 ko.T<sub>E</sub>X Live 2011 설치과정에서 별도의 설정을 하지 않았다면 \usr\texlive\2011\bin\win32로 지정해주면 된다. 다음으로 GNU 스펠체커(spell checker)인 Aspell 설치를 물어본다. 필요에 따라서 설치하도록 한다.

#### 1.2.1.3 LyX에서 한글설정

설치가 완료되었다면 LyX을 실행하고 Document → Settings → Language에서 KoreanUCS로 언어설정을 바꾸어준다.<sup>7</sup>

그리고 L<sup>A</sup>T<sub>E</sub>X Preamble에서 \usepackage\kotex\를 입력한다. Tools → Preferences → Language Settings에서 유저 인터페이스 언어와 기본 언어를 KoreanUCS로 설정하고 babel 사용을 해제하면 한글구현을 위한 기초적인 모든 작업은 완료된다. 마지막으로 이를 문서의 기본설정으로 저장하면 새로 작성하는 문서에서도 한글환경이 동일하게 구현된다.

#### 1.2.1.4 LyX에서 T<sub>E</sub>X으로

LyX에서 문서를 작성하는 경우에도 최종적인 컴파일은 ko.T<sub>E</sub>X을 사용하는 것이 편리할 수 있다. File → Export → L<sup>A</sup>T<sub>E</sub>X (pdflatex)으로 텍파일을 생성하고 TeXworks

---

<sup>7</sup>‘도은이아빠’님이 제공하는 설치파일을 사용하지 않으면 KoreanUCS가 기본적으로 제공되지 않는다.

lyx.org에서 제공하는 파일로 설치를 하여도 KoreanUCS를 추가하거나 한글을 사용할 수 있지만 기타의 세부적인 설정은 스스로 해야한다. 본인이 세부적인 설정을 직접하기를 원하는 사용자는 lyx.org에서 제공하는 원파일을 사용한다.

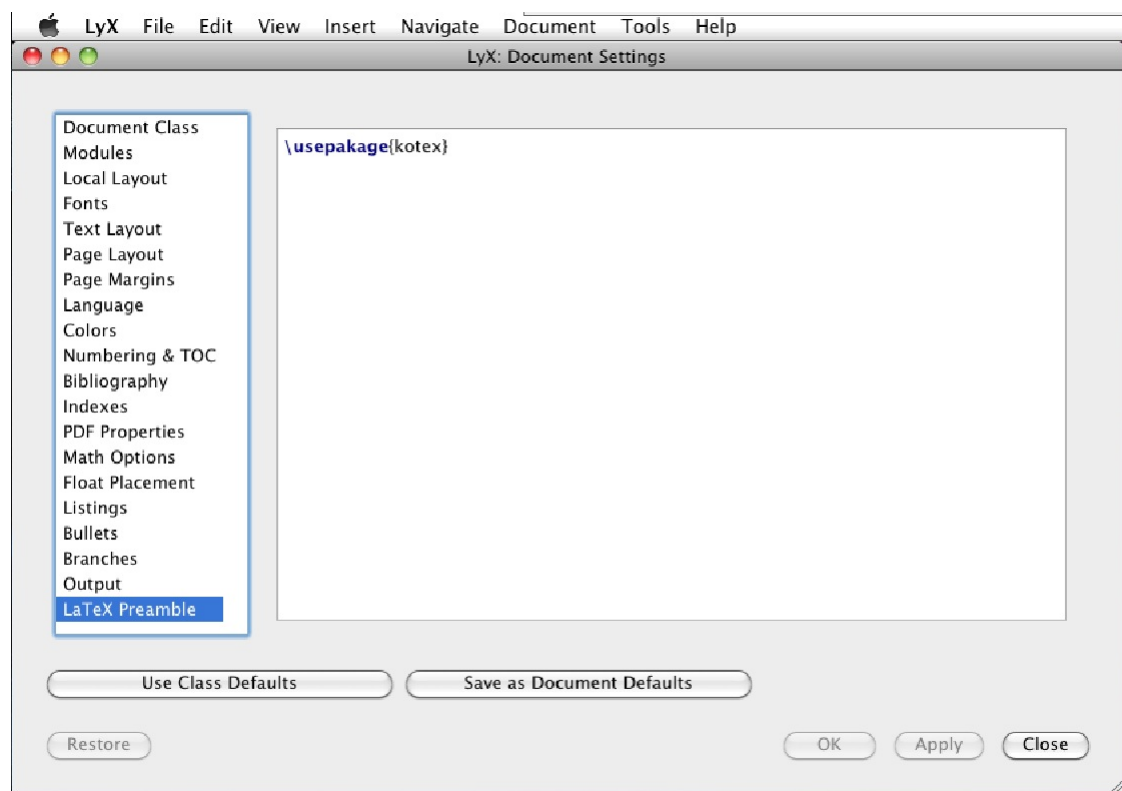


그림 1.3: LyX 한글 구현 설정화면



로 컴파일 해보면 거의 대부분의 경우 문제없이 pdf파일이 생성된다. 우리가 할 일은 프리앰블에서 필요한 부분을 손보는 것 뿐이다. 그 외에 LyX을 사용하면서 발생하는 의문사항은 Help에 있는 매뉴얼을 참조한다.

## 1.3 L<sup>A</sup>T<sub>E</sub>X시스템

### 1.3.1 L<sup>A</sup>T<sub>E</sub>X시스템에서 사용하는 파일

아래에는 L<sup>A</sup>T<sub>E</sub>X시스템에서 기본적으로 활용되는 파일을 나열하고 있다.

**.tex** : 편집기로 편집하는 문서 소스 파일

**.bib** : 사용자가 작성하는 참고문헌용 데이터베이스 파일, 3장에서 다룰 것이다.

**.bst** : 참고문헌 스타일용 파일

**.cls** : L<sup>A</sup>T<sub>E</sub>X의 문서 클래스 파일, 이에 대해서는 1장과 7 장에서 다룰 것이다.

**.def** : L<sup>A</sup>T<sub>E</sub>X 정의 파일

**.fd** : 글꼴 정의 파일

**.ist** : 인덱스 스타일 정의 파일

**.sty** : 명령 패키지의 정의 파일, 스타일

### 1.3.2 L<sup>A</sup>T<sub>E</sub>X시스템에서 생성되는 파일

**.aux** : 상호참조, 참고문헌, 목차 등에 활용되는 부수 파일

**.bbl** : BIB<sub>T</sub>E<sub>X</sub>을 사용하는 경우, 해당 문서를 위해 생성되는 참고문헌 목록 파일

**.dvi** : 화면이나 프린터 상에서 결과물을 보는 데 사용되는 파일

**.ent** : endnotes 패키지를 사용하는 경우 생성되는 파일

**.idx** : MakeIndex 명령어를 사용할 때 생성되는 파일

**.ind** : `MakeIndex` 명령어로 인덱스를 만드는 경우, 해당 문서를 위해 생성되는 인덱스 파일

**.log** :  $\text{\LaTeX}$  처리 과정에서 발생하는, 오류를 포함한 모든 정보를 담고 있는 파일

**.pdf** : 인쇄를 위한 pdf 파일

**.ps** : 인쇄를 위한 포스트스크립트 파일

**.toc** : 목차 파일, `\tableofcontents` 명령어를 사용할때 생성됨

## 1.4 $\text{\LaTeX}$ 관련 문서 및 홈페이지

### 1.4.1 관련 문서<sup>8</sup>

- 가장 기초적인 입문서들
  - $\text{\LaTeX}$  입문: The Not So short Introduction to  $\text{\LaTeX} 2_{\epsilon}$
  - Essential LaTeX
  - 워드 프로세서 사용자를 위한  $\text{\LaTeX}$
  - Basic  $\text{\LaTeX}$
  - A Beginner's Introduction to typesetting with  $\text{\LaTeX}$
  - $\text{\LaTeX}$  for Complete Novices
- 광범위한 주제를 다루고 있는 문서들
  - The UK TeX FAQ
  - $\text{\LaTeX}$  wikibook
- 논문작성을 위한 안내서
  - Using  $\text{\LaTeX}$  to Write a PhD Thesis
  - A graduate student guide to  $\text{\LaTeX}$  and AMS- $\text{\LaTeX}$

---

<sup>8</sup>아래의 모든 문서는 <http://econlecture.tistory.com/1>에서 다운 받을 수 있다.

- $\text{\LaTeX}$  による論文作成の手引き
- 기타
  - A Beamer Tutorial in Beamer
  - PostScript Drawing: An Economist's Guide
  - 한국어 텍  $\text{ko.TE\X}$  v0.1.0 사용설명서

#### 1.4.2 관련 홈페이지

- KTUG(한글  $\text{T\X}$  사용자 모임): <http://www.ktug.or.kr/>
- TUG( $\text{T\X}$  사용자 모임): <http://www.tug.org/>
- $\text{\LaTeX}$ -Wikibooks(wiki 형태의  $\text{T\X}$  백과사전): <http://en.wikibooks.org/wiki/LaTeX>
- Andrew Robert의  $\text{\LaTeX}$ 기초 가이드: <http://www.andy-roberts.net/misc/latex/index.html>
- 말레이시아의  $\text{\LaTeX}$ 기초 워크숍: <http://liantze.penguinattack.org/MMULaTeXWorkshop2010.html>
- CTAN(스타일 파일 등을 구할 때): <http://www.ctan.org/>
- $\text{\LaTeX}$  for Economics: <http://faculty.gvsu.edu/ogural/>

## 제 2 장 L<sup>A</sup>T<sub>E</sub>X 문서의 기초

앞으로 3개의 장에서는 간단한 예제를 활용하여 아주 기본적인 수준에서 논문을 작성하는 데 필요한 것을 다룰 것이다.

### 2.1 L<sup>A</sup>T<sub>E</sub>X 명령어의 기초와 컴파일

L<sup>A</sup>T<sub>E</sub>X은 조판을 위한 일종의 컴퓨터용 언어이므로, 자기 나름의 문법이 있다. L<sup>A</sup>T<sub>E</sub>X을 한번도 접해보지 않은 사람은 이 문법이 어렵게 보일 것이다. 그러나 전혀 그렇지 않다. 명령어는 일반어(영어)와 상당히 유사하다. 예를 들어 표제를 만들라는 명령 구문은 영어 그대로 ‘maketitle(make title)’이다.

명령어는 명령(command)과 환경(environment)으로 구분된다. 명령어는 일반어, 즉 본문의 내용과 구분되도록 \를 명령 구문앞에 붙인다.<sup>1</sup> 따라서 표제를 만들라는 명령은 \maketitle이다. 보통 중괄호({})안에는 명령의 적용을 받는 내용, 대괄호([])안에는 명령과 관련된 옵션을 입력한다.

환경은 \begin{명령 구문}과 \end{명령 구문}으로 구성된다. 예를 들어 문서의 시작과 끝을 지정하는 환경은 \begin{document}와 \end{document}이다<sup>2</sup>.

원고의 편집이 끝나면 결과물을 보기 위해 컴파일(compile)을 해야 한다. 컴파일은 명령어 형태로 되어 있는 원고를 우리가 보통 보는 문서 형태로 바꿔주는 과정이다. Windows의 경우 이전에는 편집기와 별도로 KCMenu를 사용하여 컴파일링을 했지만, 2010년 현재는 Texworks가 에디터와 컴파일러를 겸용한다. Mac의 경우 TeXShop이 같은 기능을 한다.

---

<sup>1</sup>문서를 작성하다 보면 명령어 및 본문 내용과 무관하게 간단한 기록을 해놓을 필요가 있다. 이 경우 %를 쓰고 그 뒤에 메모 내용을 입력한다. % 뒤의 내용은 편집기에서만 보이고, 최종 결과물에서는 보이지 않을 것이다.

<sup>2</sup>지금까지는 명확한 이해를 위해 ‘명령, 명령어, 환경’을 구분하여 썼지만 이후에서는 큰 구분없이 쓴 경우도 있을 것이다.

Texworks에서는 왼쪽 상단의 메뉴 바에 컴파일 창을 보고 원하는 컴파일을 선택한 후 왼쪽의 재생버튼을 누르면 된다. TeXShop에서는 편집 중인 파일 상단 왼쪽의 'Typeset'을 누르면 된다. 두 에디터 모두 단축기를 지원하니, 찾아 사용해보자.

## 2.2 작업 폴더와 파일 인코딩

작업 폴더의 경로에는 공백이 없는 것과 접근하기 쉬운 것이 좋다. 따라서 Windows 사용자의 경우 C:\Work처럼 루트 디렉토리에 폴더를 만드는 것을 권한다. 이는 바탕 화면이나 내 문서 같은 폴더의 경로가 C:\Documents and Settings\user로 시작하기 때문이다. Mac 사용자의 경우는 \Documents 폴더 아래 공백이 없는 이름으로 폴더를 만들어 주면 된다.

또 하나 주의할 사항은 편집기에서 파일의 인코딩을 UTF-8로 맞춰주어야 한다는 것이다. 그렇지 않으면 Windows 기본 인코딩인 CP949나 euc-kr로 저장되는데, 그러면 Mac이나 Linux 같은 다른 OS에서 한글을 읽지 못한다. 이것은 어느 플랫폼에서나 사용 가능하게 한다는 TeX의 취지에 어긋나는 것이다. 따라서 UTF-8로 항상 저장될 수 있도록 해야 한다. 현재 Windows의 주 에디터인 Texworks의 기본 코딩 설정은 UTF-8이다. 하지만, 다른 Windows 에디터는 UTF-8 저장이 기본이 아닐 수 있으므로 확인해야 할 필요가 있다. TeXShop을 사용하는 Mac 사용자는 preferences에서 Document 탭 왼쪽 하단에 있는 Encoding을 UTF-8로 지정해주면 된다.

## 2.3 프리앰블(pre-amble)

프리앰블은 본문의 시작을 지시하는 \begin{document} 이전까지 문서 전체의 형식을 지정하는 부분이다. 프리앰블에 들어간 내용이 글 전체에 영향을 주므로, 이 부분을 어떻게 작성하는가가 중요하다.

### 2.3.1 문서 클래스(document class)

```
\documentclass[옵션]{클래스 이름}
```

이 명령어는 문서를 시작하면서 문서의 성격을 정의한다. 가장 기본적인 문서 클래스는 article, report, book, slide, letter 등이다. 이외에도 특정 대학의 논문 서식처럼 특정 목적의 문서 클래스가 있다. 소속 학교의 논문 작성용 클래스가 없다면, 논문을 쓸 때는 \chapter를 사용할 수 있는 report 클래스를 사용할 것을 권한다.

```
\documentclass[10pt,a4paper,twocolumn,landscape]{report}
```

옵션은 문서 클래스를 사용자에게 맞게 세부적으로 조정하는 명령인데, 본문의 글꼴 크기, 용지 크기, 다단, 용지 가로/세로 방향 등을 조절하는데 사용한다.

이것만으로도 가장 기본적인 문서를 작성할 수 있다. 다음의 명령어를 사용해보자.

```
\documentclass[10pt,a4paper]{report}
\begin{document}
Hi, there!
\end{document}
```

### 2.3.2 패키지(package)

```
\usepackage[옵션]{패키지 이름}
```

\usepackage 명령은 좀 더 다채로운 문서 작성, 예를 들어 그림을 포함한다거나, 게임 보수표를 쉽게 그린다든지 하는 일을 쉽게 하도록 만들어 놓은 패키지를 사용하는데 사용된다. 다음의 예를 보자.

```
\usepackage[nohead]{geometry}
```

geometry는 인쇄 용지를 설정하는 패키지이며, nohead 옵션은 머리글을 사용하지 않겠다는 뜻이다.

기본적인 패키지는 이미 설치되었을 것이지만, 직접 설치해야 하는 패키지가 있다. 가장 간단한 설치법은 패키지를 사용하는 문서가 있는 폴더에 패키지 파일을 같이 두는 것이다. 모든 문서에서 사용하려면 다음의 방법을 따르면 된다. 우선 [www.ctan.org](http://www.ctan.org)에서 검색을 하여 원하는 패키지를 다운 받는다. 다운받은 패키지를 다음 위치에 복사한다.<sup>3</sup>

---

<sup>3</sup>아직, Windows에서 KCmenu 2007을 사용한다면, 자동 패키지 설치를 클릭하는 방법이 있다. 또 KCmenu의 TeX System 관리 탭을 클릭한 후, package 관리 항목의 빈 칸에서 원하는 패키지

2012년 7월 현재<sup>4</sup>, KTUG에서 ko.TeXLive 2011를 다운 받아 Windows에 설치한 경우 내장된 패키지 관리 유틸리티인 `tlmgr`을 사용하면 된다. 직접 설치하고 싶다면, `C:\usr\texlive\2011\texmf-dist\tex\latex` 폴더 아래에 하위 폴더를 만들면 된다. `geometry.sty`라면, `C:\usr\texlive\2011\texmf-dist\tex\latex\geometry\geometry.sty`로 저장하면 된다. 콘솔에 익숙한 사용자의 경우는 임의의 경로에 스타일을 넣고 `mktexlsr`를 사용하여 설치할 수 있다. TeXworks는 실행시켰을때 특정 스타일을 찾지 못하면 콘솔모드에서 정지하기 때문에 손쉽게 스타일을 적용할 수 있다.

Mac의 경우, 자기 계정에서만 사용하려면 `~/library/texmf/tex/latex/mystyles`에 설치하면 된다. 폴더가 없으면 만들면 된다.

## 2.4 한글구현

L<sup>A</sup>T<sub>E</sub>X은 서유럽 언어를 중심으로 만들어졌기 때문에 한글을 구현하기 위해서는 파일과 명령어를 추가해야 한다. 2012년 7월 현재, ko.TeX이 한글을 구현하는 통합 솔루션이다. 이 한글용 패키지는 KTUG에서 다운받을 수 있으며, 이 사이트에서 설치 프로그램을 다운받아 최초 설치한 사용자는 추가 설치를 할 필요가 없다. ko.TeX은 Unicode/UTF-8 인코딩과 EUC-KR을 모두 지원하나 원칙적으로 향후 발전 방향은 Unicode로 합의되었으므로, 본 안내서에서는 Unicode/UTF-8 인코딩만을 다룬다.

ko.TeX으로 한글을 구현하고, pdf로 만들었을 때 북마크 기능을 사용하려면 다음의 명령어를 프리앰블에 추가해야 한다.

```
\usepackage{kotex}
\usepackage[unicode]{hyperref}
```

글꼴 설치 및 자세한 한글 구현에 대해서는 ko.TeX문서를 참고하면 된다.

---

이름을 입력하고, 설치를 누르는 방법도 있다. 이 두 방법으로 설치가 되지 않는 패키지가 있으면, `\KC2007\TeX\Home\texmf\tex\latex\mystyles`에 설치하면 된다. 새로운 폴더를 만들어서 패키지를 설치했다면, L<sup>A</sup>T<sub>E</sub>X시스템에 새로운 폴더의 구조를 인식시켜주어야 한다. KCmenu에서는 TeX System 관리 탭의 갱신 항목에서 TDS 갱신, package 관리의 초기화를 누르면 된다.

<sup>4</sup>ko.TeX과 MacTeX의 폴더 트리 구조가 완전히 자리잡은 것이 아니므로 변할 수 있다.

## 2.5 표제 작성

### 2.5.1 제목, 저자, 작성일 작성

```
\title{논문 제목}
\author{논문 저자}
\date{작성일, 별도 지정하지 않을 경우, 컴파일링한 날짜가 자동 생성됨.}
\maketitle
```

여기까지가 가장 기본적인 명령어이며, 소속, 감사의 글, 저자 연락처 등을 쓰려면, 여러 가지 방법이 있겠지만 본 안내서에서는 다음과 같은 형식을 소개한다.

```
\author{저자\thanks{주소, 연락처 등}\medskip\\{\normalsize 소속}}
```

### 2.5.2 초록 작성

```
\begin{abstract}
(초록 내용 입력)
\end{abstract}
```

초록을 입력하려면 `\maketitle` 명령어 다음에 위의 명령어를 입력하면 된다.

### 2.5.3 목차 작성

L<sup>A</sup>T<sub>E</sub>X에서는 아주 쉽게 목차를 만들 수 있다. 다음의 명령어는 각각 목차, 그림 목차, 표 목차를 생성하는 명령어이다. `\maketitle` 명령어 다음에 아래 명령어를 입력하고, 두 번 컴파일링하면 된다. TeXworks나 TeXShop에서 두 번 컴파일 버튼을 눌러주면 된다.

```
\tableofcontents
\listoffigures
\listoftables
```



## 2.6 본문 작성

### 2.6.1 장, 절의 설정

장, 절을 작성하기 위한 명령어는 다음과 같다. 단 `\chapter` 명령어는 `article` 클래스에서 사용할 수 없다. 일반적으로 논문에서는 소소절(`\subsubsection`)까지 쓴다.

```
\part[목 차에 나타날 부 제목]{부 제목}
\chapter[목 차에 나타날 장 제목]{장 제목}
\section[목 차에 나타날 절 제목]{절 제목}
\subsection[목 차에 나타날 소 절 제목]{소 절 제목}
\subsubsection[목 차에 나타날 소 소 절 제목]{소 소 절 제목}
\paragraph[목 차에 나타날 단락 제목]{단락 제목}
\subparagraph[목 차에 나타날 소 단락 제목]{소 단락 제목}
```

소절의 제목까지 목차에 표시되고, 본문에도 소절의 번호까지 표시되도록 하고 싶다면, 다음의 명령어를 사용하면 된다.

```
\setcounter{tocdepth}{3}
\setcounter{secnumdepth}{3}
```

번호는 구성 순서를 지정하는 것인데, 번호와 순서는 다음과 같다.

부(part)	-1
장(chapter)	0
절(section)	1
소절(subsection)	2
소소절(subsubsection)	3
문단(paragraph)	4
소문단(subparagraph)	5

표 2.1: 장절 수준 및 번호

또 숫자를 붙이고 싶지 않지만, 장이나 절처럼 사용하고 싶으면 \*를 붙이면 된다. 용례는 다음과 같다.

```
\chapter*{장 제목}
```

이 명령어는 다른 경우, 예를 들어 수식에도 똑같이 적용될 수 있다.

이렇게 하면 목차에 장이 나타나지 않을 것이다. 이 경우 다음의 명령어를 사용하면 된다.

```
\setcounter{chapter}{장번호 재설정}
```

```
\chapter*{장 제목}
```

```
\addcontentsline{toc}{chapter}{*가 붙은 장 제목}
```

## 2.6.2 본문

다음의 구두점은 별도의 명령어를 사용해야 한다.

**따옴표:** ‘ ’, ‘ ‘ ’’, 여는 따옴표는 표준 자판의 가장 윗줄 제일 왼쪽키(숫자 1 왼편)에 할당되어 있는 문자를, 닫는 따옴표는 보통 사용하던 엔터/리턴키 왼편의 키를 사용한다.

**틸데:** `\~{}`를 사용하면 ‘`~`’와 같이 나온다. `$_sim$`을 사용하면 ‘`~`’와 같이 나온다.

**하이픈:** `abstra-ct` → ‘`abstra-ct`’ (프리앰블에 `\sloppy`를 입력하면, 자동으로 영어 하이픈을 처리한다.)

**엔대시:** `abstra--ct` → ‘`abstra-ct`’

**엠대시:** `abstra---ct` → ‘`abstra—ct`’

**말줄임표:** `$_cdots\cdots$`와 같이 입력하면 `.....`와 같이 나온다. 영어의 경우 `\ldots`로 입력하면 `...`으로 나온다.

**기타:** `\#, \$, \%, \&, \_, \{ \}, \^{} ,$_backslash$`와 같이 입력하면 `#, $, %, &, _ , { }, ^, \`과 같이 나온다.

### 2.6.3 각주와 난외주

`\footnote{각주 내용}`

각주는 위 명령어를 사용하여<sup>5</sup> 입력한다. 용례는 다음과 같다.

각주는 위 명령어를 사용하여`\footnote{각주}` 입력한다.

제목에 각주를 달아야하는 경우가 있다. 이런 경우 다음과 같이 쓴다. 반드시 [목차에 나탁날 절 제목] 옵션을 사용하여, 목차에 각주가 나타나지 않도록 해야 한다.

`\section[목차에 나탁날 절 제목]{절 제목\protect\footnote{각주 내용}}`

표 안의 각주에 대해서는 4장에서 다룰 것이다.

`\marginpar{난외주 내용}`

난외주

문서의 왼쪽이나 오른쪽 여백에 간단한 메모, 원문의 페이지, 코멘트 등을 기록하고 싶을 때가 있다. 이때는 `\marginpar` 명령어를 사용한다. 문서의 오른쪽에 나타나는 것이 기본값이다. 왼쪽에 나타나게 하고 싶다면, 프리앰블에 `\reversemarginpar` 명령어를 입력한다. 용례는 다음과 같다.

문서의 오른쪽에 나타나는 것이 기본값이다`\marginpar{난외주}`.

### 2.6.4 인용

인용문을 입력하는 명령어는 다음과 같다. 다음과 같이 쓰면 양쪽 들여쓰기로 자동 정리되어 본문에 나타난다.

`\begin{quote}`

(인용문 내용)

`\end{quote}`

### 2.6.5 참고문헌, 표, 그림 넣기

참고문헌은 3장, 표와 그림 넣기는 4장에서 다룰 것이다.

---

<sup>5</sup>각주

### 2.6.6 상호참조

논문을 쓰다 보면, ‘이 내용은 1.1 절을 참고하십시오,’ ‘수식 2.2에 따르면’ 등의 표현을 사용할 때가 있다. 이럴 때 사용하는 것이 상호참조 기능이다. 상호참조 명령어는 세 가지이다. `\label{참조 기호}`, `\ref{참조 할 기호}`, `\pageref{참조 할 기호}`. `\label`은 참조 대상이 되는 부분을 지정하고, `\ref`는 참조 대상의 번호 등을 불러오고, `\pageref`는 참조 대상의 쪽 번호를 불러온다. 참조 기호는 보통 `prefix:suffix`의 형태로 쓰지만, 각자 알아서 입력하면 된다. `prefix`는 장(chapter)은 `cha`, 절(section)은 `sec`, 수식(equation)은 `eq`, 그림(figure)은 `fig`, 표(table)은 `tab`을 주로 쓴다. 다음의 예제를 보자.

```
\subsection{상호 참조}\label{sec:cr}
```

상호 참조에 관한 내용은 `\ref{sec:cr}` 절, `\pageref{sec:cr}` 쪽에 있다.

### 2.6.7 프로그램 코드를 그대로 입력하기

본문이나 부록에 자신이 짠 프로그램 코드를 입력하고자 할 경우가 있다. 이때는 `verbatim` 명령어를 사용한다.

```
\begin{verbatim}
```

(프로그래밍 코드)

```
\end{verbatim}
```

만약 파일로 입력하고자 한다면, 프리앰블에 `\usepackage{verbatim}`을 입력하고, 다음의 명령어를 사용한다.

```
\verbatiminput{파일 이름}
```

이외에도 `listings` 패키지를 사용하는 방법도 있으나, 이는 관련 안내서를 참고하기 바란다.

### 2.6.8 각 장을 파일별로 나누기

논문을 작성하는 처음에는 목차를 변경하기 쉽도록 한 파일에서 입력하면 좋지만, 논문이 어느 정도 궤도에 오르면 각 장의 길이가 길어지므로, 하나의 파일에 관리하기

보다 각 장별로 파일을 만드는 것이 편하다. 이럴 경우 프리앰블과 표제를 만들어 놓은 파일에 다음 명령어를 입력한다.

```
\include{경로 및 파일이름 지정, 예를 들어 ch_1.tex}
```

그리고 ch\_1.tex 파일은 프리앰블 없이 chapter{장 제목}나 section{절 제목}으로 시작하고, 장의 내용을 입력하면 된다.

만약 특정 장만 컴파일링하고 싶다면, 프리앰블에 다음 명령어를 입력한다.

```
\includeonly{파일이름, 파일이름}
```

## 2.7 개조식 문서 작성

### 2.7.1 개조식 문서 작성을 위한 명령어

논문을 쓰다보면 문장부터 시작하기 보다는 마인드 맵을 그리는 것처럼 각각의 주제를 잡고, 그에 대한 부연 설명을 첨부시켜 나가는 것이 편할 수도 있다. 이런 경우 보통 찍땡식이라고 하는 개조식 문서로 쓰는 것이 생각을 정리하는 데 더 도움이 될 수 있으며, L<sup>A</sup>T<sub>E</sub>X은 이것을 쉽게 처리할 수 있다. 또 이러한 표현은 공공기관의 문서와 유사한 형태를 갖는다.

기호로 나열하는 경우 itemize 명령 구문을 사용하고,

```
\begin{itemize}
\item 제 1수준, 첫번째 항목
  \begin{itemize}
\item 제 2수준, 첫번째 항목
  \begin{itemize}
\item 제 3수준, 첫번째 항목
  \begin{itemize}
\item 제 4수준, 첫번째 항목
  \end{itemize}
\end{itemize}
\end{itemize}
\end{itemize}
```

```

\end{itemize}
\item 제 1수 준, 두 번째 항목
\end{itemize}

```

숫자로 나열하는 경우, `enumerate` 명령 구문을 사용한다.

```

\begin{enumerate}
\item 제 1수 준, 첫 번째 항목
\begin{enumerate}
\item 제 2수 준, 첫 번째 항목
\begin{enumerate}
\item 제 3수 준, 첫 번째 항목
\begin{enumerate}
\item 제 4수 준, 첫 번째 항목
\end{enumerate}
\end{enumerate}
\end{enumerate}
\end{enumerate}
\item 제 1수 준, 두 번째 항목
\end{enumerate}

```

그외, 주제어에 대한 설명만 바로 쓰고 싶다면 `description` 구문을 사용하면 된다.

```

\begin{description}
\item[제 1수 준] 첫 번째 항목
\begin{description}
\item[제 2수 준] 첫 번째 항목
\begin{description}
\item[제 3수 준] 첫 번째 항목
\begin{description}
\item[제 4수 준] 첫 번째 항목
\end{description}
\end{description}
\end{description}
\end{description}

```

```

\end{description}
\item[제 1수 준] 두 번째 항목
\end{description}

```

## 2.7.2 모양 바꾸기

앞머리에 붙는 기호를 바꾸기 위해서는 `\item` 명령의 옵션인자를 지정하거나,

```

\begin{itemize}
\item[?] 물음표로 시작하는 항목
\end{itemize}

```

프리앰블에 `\usepackage{enumerate}`를 입력하고, `enumerate` 환경의 옵션인자에 `I.`, `i.`, `A.`, `a.` 옵션을 주면 각각 `I`(로마자 대문자), `i`(로마자 소문자), `A`, `a` 로 시작하게 된다. 만약 옵션 인자에 텍스트를 쓰고자 할 때는 중괄호로 묶어주어야 한다.

```

\begin{enumerate}[문 제] I.]
\item 다음을 간략히 서략.
  \begin{enumerate}[i.]
\item 한 경제가 다음과 같은 상황에 있다.
  \begin{enumerate}[A.]
\item 국민소득  $\sim$ 
  \begin{enumerate}[a.]
\item 투자  $\sim$ 
\item 저축  $\sim$ 
  \end{enumerate}
  \end{enumerate}
  \end{enumerate}
\end{enumerate}
\item 다음을 간략히 서략.
\item 다음을 간략히 서략.
\end{enumerate}

```

## 제 3 장 참고문헌 작성

### 3.1 BibTeX의 개요

L<sup>A</sup>T<sub>E</sub>X에서는 기본적으로 thebibliography 환경<sup>1</sup>을 이용하여 참고문헌 목록을 작성할 수 있다. 그러나 논문을 작성할 때 그리고 앞으로 계속 논문을 쓴다면 BibTeX을 이용하여 참고문헌용 데이터베이스를 만들고 이를 활용하여 참고문헌 목록을 만들 것을 강력히 권한다. BibTeX의 장점은 다음과 같다.

1. 본문에 언급된 참고문헌만 정확하게 참고문헌 목록에 포함시킬 수 있다.
2. 참고문헌을 데이터베이스로 관리할 수 있다.
3. 인용 순서나 알파벳 순서 등으로 참고문헌을 정리하여 보여주기 쉽다.
4. 스타일 파일을 사용하면 참고문헌을 보여주는 형식을 바꾸기도 쉽다.

BibTeX을 사용하기 위해 우선 참고문헌 데이터베이스를 만들어야 한다.

### 3.2 참고문헌 데이터베이스 만들기

참고문헌 데이터베이스 파일의 확장자는 ‘.bib’이다. 이는 텍스트 파일이므로 어느 편집기에서도 편집이 가능하다.

우선 지금 사용중인 L<sup>A</sup>T<sub>E</sub>X 편집기를 실행하고, library.bib 파일을 만들자. 패키지 파일과 마찬가지로, 사용하려는 문서와 같은 폴더에 .bib 파일을 두어도 되지만, 어느 문서에서나 항상 사용하려면, Windows의 경우, C:\usr\texlive\texmf-local\bibtex\폴더에 두면 된다.<sup>2</sup> Mac의 경우 ~library/texmf/bibtex/bib 폴더에 파일을 두면 된다.

참고문헌 데이터베이스는 일반적인 데이터베이스와 마찬가지로 정해진 필드에 값을 입력함으로써 만들 수 있다. 기본적인 형식은 다음과 같다.

---

<sup>1</sup> 이에 대해서는 기본 안내서들을 참고하면 된다.

<sup>2</sup> 아직 KC2007을 사용한다면, \KC2007\TeX\Home\texmf\bibtex\bib 폴더에 저장하면 된다.



```
@참고문헌 종류{키워드,
  필드 이름="필드값",
}
```

기본적인 참고문헌 종류는 이미 정해져 있으며 그 내용은 표 3.1에서 확인할 수 있다. 필요에 따라 홈페이지 주소, 판결문 등의 종류를 직접 추가할 수도 있다.

article	일반 논문
book	책
booklet	출판사가 없이 인쇄된 저작
proceedings	학술대회 발표문 모음집
conference	inproceedings와 동일
inbook	어떤 책의 부, 장, 또는 절 등
incollection	어떤 책의 한 장이지만 독자적인 저자와 제목이 있음
inproceedings	학술대회 발표문 모음집 중의 발표문 하나
manual	기술문서
masterthesis	석사학위 논문
phdthesis	박사학위 논문
techreport	기관에서 출간한 보고서
unpublished	저자와 제목이 있는 미출간 원고, mimeo 등에 쓰면 됨

표 3.1: 표준 BibTeX 참고문헌 종류 입력값

키워드는 인용을 할 때 참조값으로 불러오기 위해 필요하다. 참고문헌 인용은 상호 참조 만드는 방법과 동일하다.

예를 들어, 어떤 참고문헌을 인용한다고 할 때 이 참고문헌에 Lee:1994lx 라는 키워드를 주고, 본문에서 `\cite{Lee:1994lx}`라고 인용명령을 입력하면, 해당 문헌을 인용하게 된다. 구체적인 인용방법은 다음 절에서 다룰 것이다.

키워드는 영문과 숫자만 가능하다. 키워드는 저작의 성:참고문헌 발간년도+식별

코드<sup>3</sup>로 만드는 것이 일반적이다. 인용방법은 다음 절에서 다룰 것이다.

참고문헌 종류에 따라 필수적으로 입력해야 하는 필드는 다음과 같다.

---

article	author, title, journal, year
book	author or editor, title, publisher, year
booklet	title
proceedings	title, year
inbook	author or editor, chapter or pages, title, publisher, year
incollection	author, title, booktitle, publisher, year
inproceedings	author, title, booktitle, year
manual	title
masterthesis	author, title, school, year
phdthesis	author, title, school, year
techreport	author, title, institution, year
unpublished	author, title, note

---

표 3.2: 표준 BibTeX 참고문헌 종류별 필수 필드

그외 volume, number, month, pages, address, edition, type 등의 필드값을 참고문헌 종류에 따라 부가적으로 입력할 수 있다.

저자가 여러 명일 경우 and를 사용하여 구분한다. 단, 한글 문헌 저자가 여러명인 경우 and를 사용하면, 참고문헌 목록에서 마지막 저자 앞에 and가 붙는다. 이것을 없애려면, 저자 모두를 쉼표로 쓰고 중괄호로 한 번 더 묶어준다. 즉 다수의 저자를 하나의 필드값으로 입력하면 된다. 다음 예제에서 용례를 확인할 수 있다.

다음은 참고문헌 데이터베이스 작성의 용례이며, 중괄호({}) 대신 겹따옴표(“”)를 사용해도 된다.

```
@techreport{Min:2006vn,
```

---

<sup>3</sup>식별코드는 동일년도에 동일 저자가 여러 개의 저술을 냈을 경우 구분하기 위해 필요하다.

```

    Author = {{민승규, 김용기, 이갑수, 김근영, 손민중, 양준호, 최희
갑, 최홍, 심창섭}},
    Institution = {삼성경제연구소},
    Month = {8월},
    Title = {소득양극화의 현황과 원인},
    Type = {연구보고서},
    Year = {2006},
}

@article{Berninghaus:2003ve,
Author = {Siegfried Berninghaus and Werner G{\u}th and Hartmut Kliemt},
Journal = {Journal of Evolutionary Economics},
Month = {Oct.},
Number = {4},
Pages = {385-410},
Title = {From teleology to evolution},
Volume = {13},
Year = {2003},
}

```

자신이 갖고 있는 모든 참고문헌 목록을 일일이 입력해도 되지만, 그보다 문헌 관련 사이트에서 지원하는 참고문헌 추출 기능을 활용하는 것이 훨씬 편하다. <http://ideas.repec.org/> 이나 <http://www.ssrn.com/><sup>4</sup>은 이러한 기능을 지원한다. <http://www.jstor.org/>도 이 기능을 지원한다.

### 3.3 인용 방법 및 참고문헌 목록 스타일

L<sup>A</sup>T<sub>E</sub>X에 내장된 참고문헌용 패키지도 좋지만, natbib 패키지가 그 보다 더 강력한 능력을 갖고 있다. ko.T<sub>E</sub>X에 기본 내장되어 있지만, 별도로 다운받아야 한다면, <http://tug.ctan.org/tex-archive/macros/latex/contrib/natbib/>에서 찾을 수 있다. 패

---

<sup>4</sup>회원가입이 필요하다.

키지의 옵션을 확인하려면 이 주소에서 관련 문서를 다운받으면 된다. 참고문헌을 정리하려면, natbib 패키지를 프리앰블에서 지정하고, 사용할 참고문헌 스타일과 데이터 베이스를 본문의 마지막에 지정한다.

```
\usepackage[sort&compress,comma,longnamesfirst]{natbib}
...
\begin{document}
...
\bibliographystyle{참고문헌표현스타일파일이름}
\bibliography{참고문헌데이터베이스파일이름}
\end{document}
```

스타일 파일을 사용하면 참고문헌 목록과 인용방식을 깔끔하게 정리할 수 있다. 기본적인 스타일은 다음과 같다.

---

abbrv	알파벳 순 정렬, 저자 이름 축약
alpha	알파벳 순 정렬, 참고문헌 목록에 저자 성과 출판 년도가 표시됨
plain	알파벳 순 정렬, 참고문헌 목록에 번호 붙음
unsrt	인용 순 정렬, 참고문헌 목록에 번호 붙음

---

표 3.3: 기본 참고문헌 스타일

기본 스타일에 추가하여 경제학에서 사용하는 참고문헌 인용방식 및 참고문헌 목록 작성용 스타일을 사용하려면 다음 주소에서 다운 받으면 된다. <http://www.ctan.org/tex-archive/biblio/bibtex/contrib/economic/>. 이 스타일 중에는 natbib 패키지를 필수로 설치해야 하는 것이 있다.

여기서 받은 파일 중 확장자가 .sty, .cls인 파일은 15쪽에서 지정한 것처럼 개인 스타일용 폴더로 옮겨 놓아야 한다. 확장자가 .bst인 파일은 Windows의 경우, C:\usr\texlive\texmf-local\bibtex\ 폴더에 두면 된다<sup>5</sup>. Mac의 경우 ~local/library/texmf/bibtex/bst 에 옮겨 놓아야 한다. 폴더가 없으면 직접 만들면 된다.

<sup>5</sup>아직 KC2007을 사용하는 경우, \KC2007\TeX\Home\texmf\bibtex\bst 에 저장하면 된다.

LyX에서 Natbib과 Jurabib의 사용은 Document → Settings → Bibliography에서 설정한다.

한글 참고문헌이 포함되어 있는 경우 결과물이 깔끔하게 나오는 *ecca* 스타일 사용을 권한다. 또는 Osborne 교수가 만든 *te* 스타일의 결과물도 깔끔하다. 아직 일반적이지는 않지만, 각 저널별로 자신들의 형식에 맞는 스타일 파일을 만들어 놓기도 했다. 해당 저널 홈페이지에서 직접 확인하면 된다.

인용을 하려면 다음의 명령어를 사용한다.

이 안내서는 `\cite{Talbot:2004aa}`를 기본으로 만들어졌다.

인용형식에 따라 명령어가 다른데, *natbib* 패키지와 *ecca* 스타일의 조합을 사용하는 경우 그 용례<sup>6</sup>는 다음과 같다.

---

<code>\cite{키워드}</code>	저자명 (출판연도)
<code>\cite[pp. 43-4]{키워드}</code>	(저자명, 출판연도, pp. 43-4)
<code>\citep{키워드}</code>	(저자명, 출판연도)
<code>\citeauthor{키워드}</code>	저자명
<code>\citeyear{키워드}</code>	출판연도
<code>\cite{키워드, 키워드}</code>	저자명 (출판연도), 저자명 (출판연도)
<code>\nocite{키워드}</code>	본문에는 인용이 없으며, 참고문헌 목록에만 나타남

---

표 3.4: 인용 명령어 용례

참고문헌의 결과물을 볼 차례이다. BibTeX의 사용 순서는 다음과 같다.

1. 참고문헌 데이터베이스 만들기
2. 본문에 인용 문헌을 지정

---

<sup>6</sup>다른 패키지와 스타일의 조합을 사용하더라도 명령어는 동일하지만, 패키지와 옵션, 스타일의 지원 여부에 따라 인용형식의 모양이 변할 것이다. 현재의 예는 *comma* 옵션을 적용한 결과이다. 콜론으로 인용 문헌을 구분하고 싶다면 *comma* 옵션을 지우면 된다.

### 3. BibTeX 컴파일링<sup>7</sup>

### 4. L<sup>A</sup>T<sub>E</sub>X 컴파일링 두 번<sup>8</sup>

TeXworks나 TeXShop의 BIBTeX 컴파일링은 ‘typeset’ 버튼 왼쪽의 메뉴에서 BiBTeX을 선택하면 된다.

참고로 unsrt 기본 스타일과 \nocite 명령어를 사용하면, 논문 준비중 reading list를 만드는 데 편리하다.

#### 3.3.1 참고문헌 관리 프로그램

bib 파일은 텍스트로 저장되므로 어디에서든 열어볼 수 있는 장점이 있다. 하지만, 정리해서 보여주는 것이 아니므로 보기와 찾기에 안 좋다. 참고문헌 관리에 특화된, 즉 bib 파일용 에디터로는 Mac의 경우 Bibdesk(오픈소스)와 Papers(유료)를 손꼽을 수 있다.

Windows의 경우, BibTeX을 직접 관리하는 프로그램으로 Java에 기반한 JabRef가 있다. 하지만 서지관리프로그램의 대부분은 BibTeX 형식의 입출력을 지원하기 때문에 굳이 이를 사용할 필요성은 없다. Endnote, RefWorks, Mendeley 등의 서지관리프로그램이 BibTeX 형식의 입출력을 지원한다. 서지관리프로그램은 정답이 없기 때문에 스스로 제일 편리하다고 생각되는 프로그램을 사용할 것을 추천한다.

학교 등에서 Endnote를 다운 받아 설치 할 수 있으면, Endnote 파일을 BibTeX 파일로 전환할 수도 있다. 참고문헌 관리를 위한 프로그램 목록은 <http://dret.net/projects/sharef/related-projects>에서 볼 수 있다.

여기서는 Mendeley를 소개하도록 한다. Mendeley는 일정 한도내에서 무료이고 서로 다른 OS 환경에서 동일한 서지환경을 제공한다는 장점이 있다.<sup>9</sup> Mendeley를 설치하면 File → export에서 서지를 BibTeX 형식으로 출력할 수 있다. 만들어진 .bib 파일은 앞에서 설명한 바와 같이 사용할 수 있다.

---

<sup>7</sup> 해당 문서의 인용과 데이터베이스를 대조하여, 해당 문서에 해당하는 참고문헌 목록을 별도 파일(문서파일명.bbl)로 만든다.

<sup>8</sup> 본문의 인용 부분에 정확한 인용 결과가 표시된다.

<sup>9</sup> 우분투와 윈도우즈, OSX과 윈도우즈에서 동일한 서지관리환경을 구축할 수 있다는 말이다.

## 제 4 장 표와 그림

4장에서는 표와 그림을 다루는 방법에 대해 알아본다. 우선 간단한 그림을 하나 넣어보면서  $\text{\TeX}$ 에서 그림을 넣는 법<sup>1</sup>에 대해서 알아보도록 하자.

### 4.1 그림 넣기

#### 4.1.1 예제 1 : 간단한 그림 넣기

.jpeg나 .jpg 확장자의 그림을 넣기 위해서는 프리앰블에 `graphicx` 패키지의 사용을 지정한다.

```
\usepackage{graphicx}
```

여기서는, <https://dl.dropbox.com/u/2914888/tex/cat.jpg>에서 사진을 다운받고,  $\text{\TeX}$ 연습용으로 쓰고 있는 폴더에 넣어두자. 그리고 아래와 같이 입력해본다.

```
\begin{figure}[htbp] %그림 시작, 그림의 위치 옵션을 htbp로 설정
\begin{center} %가운데 정렬
\includegraphics[width=0.2\textwidth]{cat.jpg} %그림 파일 입력
\caption{\TeX \이 뭐야? 먹는 거야?} %그림 설명 입력
\label{fig:cat} %참조용 라벨
\end{center}
\end{figure}
```

그림이 제대로 들어가지 않고 에러가 난다면, 그림의 크기(bb: bounding box)를 잡아주어야 한다. `bb=0 0 600 600`은 그림의 크기를 지정하는 것인데, 첫번째와 세번째는 가

---

<sup>1</sup> 좀 더 엄밀하게 표현하자면, 떠다니는 개체(Floating Object)를 넣는 법이라고 해야 한다.  $\text{\TeX}$ 에서는 모든 객체가 일종의 상자(box)로 취급된다. 가장 작은 상자는 바로 글자이다. 표나 그림은 커다란 상자라고 보면 되겠다. 이러한 상자들을 통칭하여 떠다니는 개체라고 하며, 이것의 배치법은 일정한 규칙에 따른다.

로 크기, 두번째와 네번째는 세로 크기를 지정한다. 예를 들어 그림의 크기가 1024X768 이라면, `bb=0 0 1024 768`이 된다.



그림 4.1: T<sub>E</sub>X이 뭐야? 먹는거야?

아마도 그림은 원하는 위치에 오지 않았을 것이다. T<sub>E</sub>X는 기본적으로 주어진 위치 옵션에 따라 자동 배치를 한다<sup>2</sup>. 위 예제 중 그 옵션 지정에 해당되는 부분만 남기면 아래와 같다.

```
\begin{figure}[htbp]
...
\end{figure}
```

htbp 옵션은 각각 h는 바로 이 자리(here), t는 문서의 최상단(top), b는 문서의 최하단(bottom), p는 별도의 그림만 모아놓은 장(page)를 의미한다.

T<sub>E</sub>X는 미리 정의된, 떠다니는 개체의 크기에 따른 배치 조건이 설정되어 있으며, 첫 조건이 적용 불가능할 경우 우선순위에 따라 다음 조건의 적용 가능성을 탐색해 나간다. 모든 탐색조건을 충족하지 못한다면(디폴트), 그 경우 p 옵션이 발동되어 떠다니는 개체만을 모은 페이지로 가게 된다. 만일 이러한 디폴트 조건을 무시하고 싶다면 옵션 뒤에 ‘!’를 붙이면 된다. 가령, hwp의 ‘글자처럼 취급’ 옵션처럼 절대 조건과 상관없이 그 자리로 가게 하고 싶을 경우 [h!] 옵션을 사용한다. 아무 옵션도 붙이지 않으면 [tbp]로 설정된다. 만일 hwp의 ‘글자처럼 취급’ 옵션과 완전히 동일한 방식을 쓰고 싶다면, 넣고 싶은 부분에 `includegraphics` 명령어를 넣으면 된다. 이렇게 말이다.

```
\includegraphics[width=0.2\textwidth]{cat.jpg}
```

---

<sup>2</sup>물론 수동으로 정확하게 자신이 의도한 자리에 의도한 배치를 하게 할 수는 있지만, 그것은 직접 기술 문서를 참조하면 알 수 있다.



위 명령어는 들어갈 그림파일의 위치를 명시하고 있다. 파일 이름의 확장자는 써도 되지만, 명시하지 않아도 무방하다.  $\text{\TeX}$ 는 파일명 뒤에 자신이 지원하는 그림 파일 확장자를 붙여가며 탐색을 하기 때문이다. 대괄호 안의 `width=0.2\textwidth`는 폭을 문서 가로폭의 20%로 하겠음을 명시하는 것이다. 당연히 0.2를 가령 0.5로 만들면 폭의 50%만큼의 그림이 되는 것이다. 절대폭으로 만들고 싶다면 `width=5in`, `width=7cm`와 같은 식으로 단위와 함께 명시할 수 있다. 폭보다 높이가 중요하다면 `width` 대신 `height`를 쓰면 되며, 페이지 높이에 대한 비율을 사용하기 위해서는 `\textheight`를 사용하면 된다.

```
\caption{\TeX \이 뭐야? 먹는 거야?}
```

`\caption` 명령어는 그림에 붙일 캡션을 설정한다. 별도의 위치 지정이 없으면 그림 아래에 만들어진다. 표에서도 같은 방식으로 사용된다. `\TeX`는  $\text{\TeX}$ 을 표현하기 위한 명령어이고, `\`은 `kotex package`를 설치하면 사용할 수 있는 명령어로, 앞에 붙는 말에 따라 조사 ‘은’, ‘는’ 중에 적당한 것을 붙여준다. 물론 ‘이’, ‘가’ 앞에 역슬래시(`\`)를 붙여도 된다. 단, 이 기능을 사용하는 경우, `dvi`로 컴파일할 때 목차가 깨져 나올 수 있다. `dvi`로 컴파일을 하는 경우는 본 안내서의 수준에서는 `PSTricks`를 사용할 때 뿐이며, 이에 대해서는 9장에서 다시 다룬다.

상호참조에 관해서는 20쪽을 참조하라.

다음 명령어는 상대크기를 축소하고, 그림을 회전시키며, `!`를 사용하여 위치를 지정한 응용 예이다.

```
\begin{figure}[h!]
  \begin{center}
    \includegraphics[angle=45, width=0.5\textwidth]{cat.jpg}
    \caption{상대크기를 폭의 50\%로 축소하면서 반시계방향으로 45도 꺾음}
    \label{fig:cat_angle}
  \end{center}
\end{figure}
```



그림 4.2: 상대크기를 폭의 50%로 축소하면서 반시계방향으로 45도 꺾음

## 4.2 표 만들기

표는 table 환경에 tabular환경을 넣어 사용한다. 실제 표를 만드는 것은 tabular 환경이고, table환경은 표의 위치지정 옵션 및 참조레이블, 캡션, 표 번호 등을 관리하는 데에 쓰인다. 사실 그림 역시 figure환경에 includegraphics 명령을 사용하여 그림을 넣은 것이다. figure환경 없이 includegraphics 명령만을 사용해도 그림은 들어간다.

### 4.2.1 간단한 표의 예

```
\begin{tabular}{|c||c|c|}
\hline
& 자백 & 부인 \\
\hline \hline
자백 & 5,5 & 1,100 \\
\hline
부인 & 100,1 & 0,0 \\
\hline
```

`\end{tabular}`

위 명령어를 실행시키면 보는 것과 같은 결과를 얻을 것이다. 이는 그림과 유사하지만, 위치 지정자(htbp)나 가운데 정렬방식 등을 설정하지 않았으므로 거대한 한 글자와 같이 취급되고 있다. 그래서 부자연스럽게 같은 줄에 위치하고 있다. 이는 hwp의 ‘글자와 같이 취급’옵션과 완전히 동일한 것이다. 그림 역시 table 환경 없이 includegraphics

명령만 사용할 경우 같은 효과를 거둘 수 있다.

	자백	부인
자백	5,5	1,100
부인	100,1	0,0

위 표 대신 정식으로 table 환경을 적용한 경우는 아래와 같다.

```
\begin{table}[h!]
\begin{center}
\begin{tabular}{|c||c|c|}
\hline
& 자백 & 부인 \\
\hline
\hline
자백 & 5,5 & 1,100 \\
\hline
부인 & 100,1 & 0,0 \\
\hline
\end{tabular}
\end{center}
\caption{가운데 정렬한 표}
\end{table}
```

표의 내용을 채우고 있는 부분은 수식의 행렬과 매우 유사한데, 행 구분은 `\\`으로, 열 구분은 `&`로 하고 있다. 주의할 점은 아무것도 없는 부분(여기에서는 1행 1열이 그러하다)이 비었다는 것을 표현하기 위해 `& 자백 & 부인 \\` 같은 식으로 표현한다는 점이다.

`\begin{tabular}{|c||c|c|}`의 의미는 가운데 정렬(center)을 하는 3열짜리 표라는 것이며, 세로줄을 모두 굵게, 1열과 2열 사이의 세로줄은 두줄짜리를 쓰겠다는 것을 의미한다. 즉, 처음 표를 시작할 때 열에 대한 정보와 함께 세로선에 대한 기본 정보를 제공하는 것이라고 생각하면 된다. 가로선은 `\hline` 명령을 쓴다.

위에서 쓴 `c`대신 쓸 수 있는 것은 `l`, `r`, `p` 등이 있다. `l`은 왼쪽정렬(left), `r`은 오른쪽 정렬(right), `p`는 줄바꿈이 가능하게 한다(paragraph). 뒤에 중괄호를 붙여 크기를 강제로 지정할 수 있다. 지정하지 않는 경우  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 이 자동으로 폭을 결정해주게 된다.

## 4.2.2 조금 더 복잡한 표

좀 더 복잡한 표를 만들어 보자.

### 4.2.2.1 열병합 표

열 병합은 `multicolumn` 명령을 사용한다. 표 4.1의 미완성된 줄(2.1과 3.1 사이)은

종 류		
C1	C2	C3
2.1	2.2	2.3
3.1	3.2	3.3

표 4.1: 열병합 표

의도적인 것이다. `\cline{}` 명령어는 가로선의 시작부분과 끝부분을 지정한다. 즉, 예에서 3번째 셀의 중간에 걸친 밑선은 2번째 셀에서 시작하여 3번째 셀에서 끝나게 하기 위해 `\cline{2-3}`을 쓴 것이다.

```
\begin{table}[!h]
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\multicolumn{3}{|c|}{종 류}\end{tabular}
\end{center}
\end{table}
```

```

\cline{1-3}
C1&C2&C3\\
\hline
2.1&2.2&2.3\\
\cline{2-3}
3.1&3.2&3.3\\
\hline
\end{tabular}
\end{center}
\caption{열 병합 표}
\end{table}

```

#### 4.2.2.2 행 병합 표

이제 감을 잡았을 것이다. 행 병합은 `multirow` 명령을 사용한다. 이때 `multirow` 명령을 사용한 셀은 빈 셀 취급을 하면 된다.

Text in column1	C2a	Text in column3	C4a
	C2b		C4b
	C2c		C4c
C1d	C2d		C4d

표 4.2: 행 병합 표

```

\begin{table}[!h]
\begin{center}
\begin{tabular}{|l|l|l|l|}\hline
\multirow{3}{20mm}{Text in column1}&C2a&\multirow{4}{20mm}{Text in column3}&C4a\\
&C2b&&C4b\\
&C2c&&C4c\\
C1d&C2d&&C4d
\end{tabular}
\end{center}
\end{table}

```

```

\cline{1-2}
C1d & C2d & & C4d\\
\hline
\end{tabular}
\end{center}
\end{table}
\caption{행병합 표}

```

#### 4.2.2.3 표 속에 각주 넣기

표 안에 각주를 넣는 방법은 여러가지가 있는데, 이에 대해서는 <http://faq.ktug.or.kr/faq/Table%C8%AF%B0%E6>을 확인하고, 여기서는 `threeparttable.sty`를 사용하는 방법과 `\footnotemark`를 사용하는 방법 두 가지를 다룬다.

우선 `threeparttable.sty`을 사용하기 위해서는 해당 스타일을 설치하고, 프리앰블에 사용을 지정한다. 그리고, 다음의 예제를 보자.

```

\begin{table}
\begin{center}
\begin{threeparttable}
\begin{tabular}{|c|c|c|}
\hline
GDP\tnote{1} & GDP\tnote{2} & GDP\tnote{3}\\
\hline
\end{tabular}
{\footnotesize
\begin{tablenotes}
\item[1] 2007년 한국은행
\item[2] 2008년 한국은행 추정치
\item[3] 2008년 KDI 추정치
\end{tablenotes}
}

```

```

\end{threeparttable}
\caption{표 아래 각주를 표시하는 표}
\label{tab:footnote}
\end{center}
\end{table}

```

threeparttable 패키지의 사용에서 중요한 것은 \tnote{각주 기호}로 각주 기호를 지정하고, 이에 대한 내용은 \item[각주 기호]를 사용하여 입력한다는 것이다.

tabular 환경에서 페이지 바닥에 각주를 넣으려면, \footnotemark와 \footnotetext 명령어를 사용한다.

```

\footnotemark[번호]
\footnotetext[번호]{각주 내용}

```

용례는 다음과 같다.

```

\begin{table}[!h]
\begin{center}
\begin{tabular}{|c|c|c|}
\hline
GDP\footnotemark[1] &
GDP\footnotemark[2] &
GDP\footnotemark[3]\\
\hline
\end{tabular}
\end{center}
\caption{페이지 바닥에 각주를 표시하는 표}
\label{tab:pagefootnote}
\end{table}
\footnotetext[1]{2007년 한국은행}
\footnotetext[2]{2008년 한국은행 추정치}
\footnotetext[3]{2008년 KDI 추정치}

```

#### 4.2.2.4 Excel2Latex의 사용

지금까지 본 것처럼 L<sup>A</sup>T<sub>E</sub>X에서 표를 만들기는 쉽지 않다. 이를 좀더 편하게 해주는 유틸리티가 excel2latex이다.

이는 엑셀의 매크로 파일인데, <http://www.ctan.org/tex-archive/support/excel2latex/>에서 받을 수 있으며, Excel2LaTeX.xla 파일을 다운받아, 엑셀의 add-in 디렉토리에 설치하면 된다. 그러면 ‘도구’ 메뉴와 리본에 새로운 아이콘이 나타난 것을 볼 수 있을 것이다.

사용방법은 L<sup>A</sup>T<sub>E</sub>X의 표로 만들고 싶은 부분을 설정하고, 툴바 아이콘을 눌러주기만 하면 된다. 그러면 L<sup>A</sup>T<sub>E</sub>X으로 내용이 전환되고, copy to clipboard를 선택하면 해당 부분이 복사되어, 이를 L<sup>A</sup>T<sub>E</sub>X편집기로 옮겨 놓으면 된다.

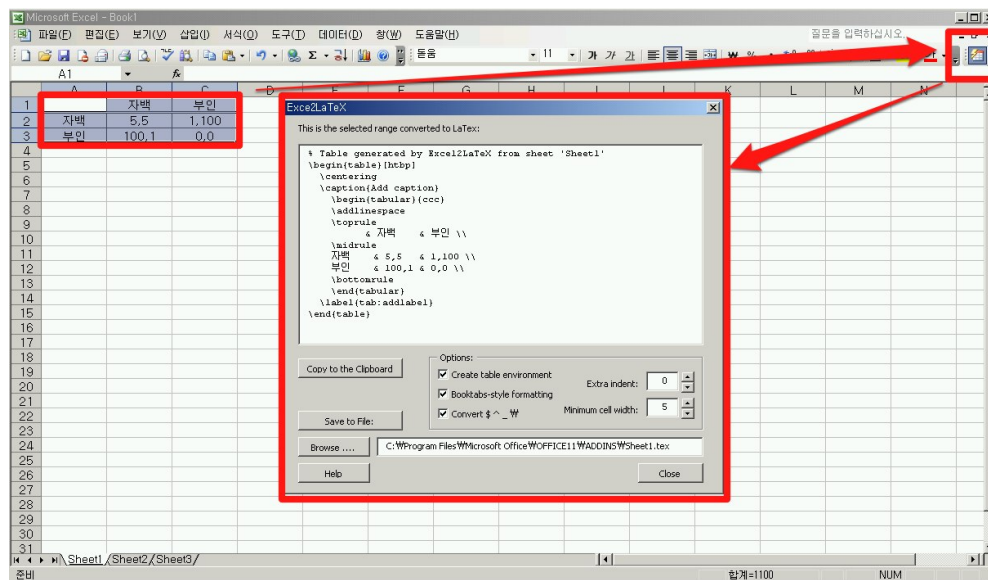


그림 4.3: Excel2Latex

#### 4.2.2.5 csvsimple 패키지의 사용

엑셀 파일을 csv(comma separate file) 확장자로 추출할 수 있다. csvsimple 패키지는 이 csv 파일을 이용하여 표 만들기를 도와준다. 먼저 프리앰블에 `\usepackage{csvsimple}`을 정의한다.



가장 기본적인 사용법은 다음과 같다.

```
\documentclass{article}

\usepackage{csvsimple}

\begin{document}
\csvautotabular{파일이름.csv}
\end{document}
```

만약, csv 파일의 데이터만 불러와서 사용하고자 한다면, `\csvreader{옵션}` 을 이용하면 된다. 자세한 내용은 관련 기술문서를 참고하자.

## 제 5 장 수식

5장에서는 수식을 다루는 법을 설명한다. 여기에서 다루는 수식은 엄밀히 말하자면  $\text{\LaTeX}$ 의 표준 방식이라기 보다는  $\text{\LaTeX}$ 이라고 할 수 있다.  $\text{\LaTeX}$ 은 미국 수학회에서 만든 수식입력을 위한 확장이다. 이 확장을 이용하기 위해서는 프리앰블에 다음과 같이 입력하자.

```
\usepackage{amssymb, amsfonts, amsmath}
```

### 5.1 수식모드

한글과 컴퓨터사에서 아래아 한글 시리즈의 수식을 써 본 사람이라면  $\text{\TeX}$ 의 수식을 훨씬 쉽게 이해할 수 있을 것이다. 사실 아래아 한글의 수식 입력법은  $\text{\TeX}$ 의 것을 차용했다고 보아도 무방하다. 다른 점은 예약어 목록 일부가 다르다는 것과 함께 아래아 한글의 경우는 그냥 예약어를 쓰는 반면,  $\text{\TeX}$ 에서는  $\backslash$ 과 함께 써야 한다는 것 정도이다.  $\text{\TeX}$ 에서 수식모드는 크게 두 가지 방법으로 쓸 수 있다.

1.  $\$$ 와  $\$$  사이에 수식을 넣는 법 (한글의 수식과 유사)
2. 별도의 수식 환경을 불러 쓰는 법

첫번째 방법은 아래아 한글과 거의 유사하게 수식을 부를 수 있다. 이것은 정렬방식을 지정하지 않은 표(아래아 한글에서는 ‘글자로 취급’ 옵션과 비슷하다)와도 비슷하게, 문장 사이에 수식을 넣을 수 있다. 가령 유명한 오일러의 공식인  $e^{\pi i} + 1 = 0$ 을 표현할 경우, 문장 내에 쓸 땐 첫번째 방법  $\$e^{\pi i}+1=0\$$ 을 쓰고, 수식 번호를 매기는 식으로 쓰려면 아래와 같이 `equation` 환경을 쓰는 것이다.

```
\begin{equation}
e^{\pi i}+1=0
\end{equation}
```

$$e^{\pi i} + 1 = 0 \quad (5.1)$$

여기에서 수식 번호는 (장번호.수식번호)의 형식으로 나타나고 있다. 이 수식을 붙이는 방식은 프리앰블에서 바꿀 수도 있고, 이 장에서만 임시로 바꿀 수도 있다.

한편, 이 두 가지 식 표현법은 정렬 방식 외에도 다른 점이 있는데, 첫번째 방식의 경우는 무조건 세로 사이즈에 상관 없이 한 줄에 들어가도록 표현한다는 것이다. 이는 적분기호나 시그마 같은 합기호를 쓸 경우 잘 보인다. 즉,  $\sum_{i=0}^{100} p_i q_i$ 와 아래의 식을 비교해 보라. 어떤 의미인지 쉽게 이해할 수 있을 것이다. 분수와 같은 경우, 크기를 줄여 표현한다.  $\frac{1}{\frac{1}{10}+1}$

$$\sum_{i=0}^{100} p_i q_i$$

$$\frac{1}{\frac{1}{10} + 1}$$

수식 모드 내에서는 일반적인 텍스트 입력과 몇 가지 면에서 다른 점이 있다.

1. 띄어쓰기는 의미가 없다. 아무리 띄어쓰기(스페이스바)를 해도 수식 내에서는 모두 무시된다. 공백은 `~`, `\quad`, `\qquad`를 사용하여 강제지정 해야 한다.
2. 수식 모드 내에서는 일반 모드 내에서는 쓸 수 없었던 예약어(수식용 예약어)들을 쓸 수 있다. 다시 말하면 수식 모드 밖에서 수식용 예약어들을 사용할 경우 컴파일 에러가 발생한다.
3. 수식 모드 내에서 일반 모드같은 조판을 하기 위해서는 특수한 예약어를 사용해야 한다.

## 5.2 수식의 표현법

### 5.2.1 기초적 용법

일반적인 다항식을 표현해보자. 아래는 2차방정식의 일반해이다.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

위 식은 아래와 같은 방법으로 표현한다.

```
\begin{equation*}
x=\frac{-b\pm\sqrt{b^2-4ac}}{2a}
\end{equation*}
```

`equation*` 환경은 숫자가 붙지 않는 완전한 수식이다.<sup>1</sup> `\frac{분자}{분모}`은 분수를 표현할 때 쓴다. 루트는 `\sqrt{}`를 쓰며, `\pm`는  $\pm$ 을, 상첨자(제곱)는 `^`으로 나타내고 있다. 중괄호 `{ }`는 묶여야만 하는 단위를 명시하는 기능이 있다. 가령  $a^2$ 는 `a^2`로도 표현할 수 있지만, `a^2`로도 표현해도 똑같은 모양으로 나타나게 된다. 하지만,  $a^{4\pi}$  같은 것은 중괄호 없이 표현하는 것이 불가능하다. 중괄호를 쓰지 않고 `a^4\pi`와 같이 쓰면  $a^4\pi$  처럼 되게 된다. 물론 이 모든 수식들은 한줄짜리 약식 수식으로 쓸 수 있다. 같은 내용을 두 개의 `\` 사이에 넣으면 된다.

### 5.2.2 첨자 있는 화살표

$$F \times \Delta[n-1] \xrightarrow[\Gamma]{\partial_0 \alpha(b)} E^{\partial_0 b}$$

상 하첨자가 들어가는 화살표는 `\xrightarrow[하첨자]{상첨자}`를 쓴다. 물론 상/하 화살표, 왼쪽으로 향하는 화살표에도 사용할 수 있다.<sup>2</sup>

```
\begin{equation*}
F\times \triangle[n-1]\xrightarrow[\Gamma]{\partial_0\alpha(b)}E^{\partial_0b}
\end{equation*}
```

### 5.2.3 적분

```
\begin{equation*}
\int_{-\infty}^{\infty}\log_e\gamma_t^2dt
\end{equation*}
```

---

<sup>1</sup>강제로 번호나 표식을 붙이고 싶다면 `\tag{ }` 명령을 쓰면 된다. 위 환경에 대한 약식 환경으로 `\[ \]` 환경을 쓰는 방법도 있다.

<sup>2</sup>원래 쓰던 `@>>>`식의 표현은 이제 사용할 수 없다.

$$\int_{-\infty}^{\infty} \log_e \gamma_t^2 dt$$

#### 5.2.4 경우의 수

줄 바꾸기 위해 4장에서 보았던 &과 \\가 사용되고 있음을 알 수 있다. 이는 이어서 살펴볼 행렬에도 똑같이 적용된다. 단, 여기에서 &는 줄을 맞출 때에만 사용하는 것이기 때문에 갯수를 맞춰야 할 필요는 없다.

$$P_{r-j} = \begin{cases} 0 & \text{if } r-j \text{ is odd,} \\ r!(-1)^{2n} & n \in \mathbb{N}. \end{cases}$$

```
\begin{equation*}
P_{r-j}=
\begin{cases}
0&\text{if } r-j \text{ is odd},\\
r!(-1)^{2n}&n\in\mathbb{N}.
\end{cases}
\end{equation*}
```

이 이외에도 여러 예약어 들이 있다. 이에 대해서는 <http://www.ams.org/publications/authors/tex/amslatex.html> 등의 사이트에서 예약어 용례와 목록 들을 찾아볼 수 있다.

#### 5.2.5 행렬

\quad, \qquad 는 수식 환경에서 띄어쓰기를 할 때 쓰는 명령어이다. gather\* 환경은 여러 개의 수식을 정렬하지 않고 모으는 데 쓰인다.

$$\begin{array}{cc} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ \left| \begin{array}{cc} a & b \\ c & d \end{array} \right| & \left\| \begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \right\| \end{array}$$

```

\begin{gather*}
\begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \quad
\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad
\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad
\begin{vmatrix} a & b \\ c & d \end{vmatrix} \quad
\begin{Vmatrix} 0 & 1 \\ 1 & 0 \end{Vmatrix}
\end{gather*}

```

### 5.2.6 align 환경

수식을 풀어나가는 과정을 묘사할 때처럼 여러 줄의 수식을 정렬해서 나타내야 할 경우, align 환경을 이용한다. 이때, 정렬해야 할 기준에 & 을 넣는다. 아래 예에서는 등호를 기준으로 정렬을 하고 있다.

$$\begin{aligned}
 ax^2 + bx + c &= 0 \\
 a\left(x + \frac{b}{2a}\right)^2 - \frac{b^2}{4a} + c &= 0 \\
 \therefore x &= \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
 \end{aligned}$$

```

\begin{align*}
ax^2+bx+c &= 0 \\
a\left(x+\frac{b}{2a}\right)^2-\frac{b^2}{4a}+c&=0 \\
\therefore x&=\frac{-b\pm\sqrt{b^2-4ac}}{2a}
\end{align*}

```

### 5.2.7 극한, 확률

$\lim$ ,  $\Pr$ 은 극한과 확률을 표시하기 위해 사용한다.  $\leq$ ,  $\geq$ 를 사용하면 부등호를 나타낼 수 있다. 앞서 살펴본 align 환경을 이용하는 긴 수식을 작성하는 예제를 살펴보자.

$$\begin{aligned}\lambda(t) &= \lim_{dt \rightarrow 0} \frac{\Pr(t \leq \tau < t + dt | \tau \geq t)}{dt} = \lim_{dt \rightarrow 0} \frac{\Pr(t \leq \tau < t + dt)}{\Pr(\tau \geq t) dt} \\ &= \lim_{dt \rightarrow 0} \frac{S(t) - S(t + dt)}{S(t) dt} \\ &= -\frac{1}{S(t)} \frac{dS(t)}{dt} = -\frac{d \log S(t)}{dt} = \frac{f(t)}{S(t)}.\end{aligned}$$

```
\begin{align*}
\lambda(t) &= \lim_{dt \to 0} \frac{\Pr(t \leq \tau < t+dt \mid \tau \geq t)}{dt} \\
&= \lim_{dt \to 0} \frac{\Pr(t \leq \tau < t + dt)}{\Pr(\tau \geq t) dt} \\
\\
&= \lim_{dt \to 0} \frac{S(t) - S(t + dt)}{S(t) dt} \\
&= -\frac{1}{S(t)} \frac{dS(t)}{dt} = -\frac{d \log S(t)}{dt} = \frac{f(t)}{S(t)}.
\end{align*}
```

### 5.2.8 LyX에서 수식입력

LyX은 TeX편집기이기 때문에 앞서 살펴본 모든 수식을 표현할 수 있다. LyX의 수식입력은 예약어 입력도 가능하고, 한글이나 MS워드처럼 마우스로 원하는 기호를 찾아서 클릭하는 방식도 가능하다.

Insert → Math를 눌러보자. ‘Inline formula’는 \$ 사이에 수식을 넣는 방식을 의미한다. ‘Display formula’는 \begin{equation\*} 또는 \[를 사용하여 수식을 넣는 방식을 의미한다. ‘Numbered formula’는 \begin{equation}을 사용하여 번호가 있는 수식을 넣는 방식이다. ‘AMS align Environment’는 \begin{align\*}을 사용하는 방식이다.

이 네 가지 수식입력 방법으로 거의 대부분의 수식은 입력이 가능하다.

LyX의 수식입력 모드에서 예약어를 사용하고 스페이스바나 커서를 누르면 작성 중인 파일을 컴파일 하지 않아도 입력한 수식을 보여준다. TeX을 처음 접하는 경우에는 아직 예약어의 사용이 익숙하지 않기 때문에 작성한 수식을 직접 보여주는 방식이 문서의 작성에 도움을 줄 수 있다. 예약어를 모르는 경우는 입력 패널에서 찾아 마우스로 클릭한다. 아래의 그림은 LyX에서 수식입력이 어떻게 이루어지는지 보여준다.

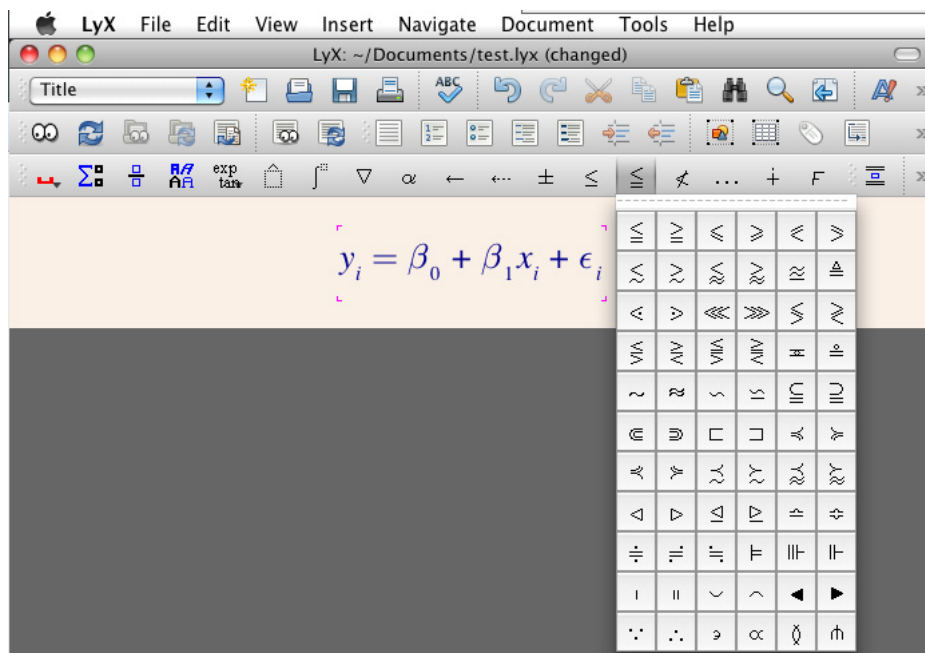


그림 5.1: LyX에서 수식입력



## 제 6 장 L<sup>A</sup>T<sub>E</sub>X 문서의 편집 기초

2장에서 익힌 것으로도 충분히, 논문을 보기 좋게 만들 수 있다. 이 장에서 소개하는 내용은 논문을 꾸미는 데 도움을 줄지도 모르지만, 필수 불가결한 것은 아니라는 점을 명심하자.

### 6.1 글자 모양

#### 6.1.1 글꼴 바꾸기

하나의 문서에 글꼴이 많이 쓰이면 쓰일 수록 통일성이 떨어지고, 한 눈에 읽기도 좋지 않다. 그러므로 되도록이면 글꼴을 유지할 것을 권한다.

그럼에도 표준 글꼴이 마음에 들지 않는다면, 다음의 명령어를 사용할 수 있다.

```
\SetHangulFonts{rm(roman)}{ss(san serif)}{tt(typewriter)}
```

```
\SetHanjaFonts{rm(roman)}{ss(san serif)}{tt(typewriter)}
```

위의 명령은 각각 한글과 한자 글꼴을 지정한다. ko.T<sub>E</sub>X에서 기본 지원하는 기본 글꼴 목록과 기본 설정은 표 6.1, 표 6.2와 같다. 확장 글꼴과 트루타입 글꼴의 추가와 같이 더 자세한 사항은 ko.T<sub>E</sub>X가이드 문서를 참조하라.

글꼴	글꼴 이름
명조	utbt
고딕	utgt
타자	uttz
그래픽	utgr

표 6.1: ko.T<sub>E</sub>X 기본 글꼴 목록

언어 종류	rm 글꼴 이름	sf 글꼴 이름	tt 글꼴 이름	emph 글꼴 이름
한글	utbt	utgt	uttz	utgr
한자	utgt	utgt	uttz	utgr

표 6.2: ko.T<sub>E</sub>X문서 한글 기본 설정

### 6.1.2 글꼴 크기

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

표준 크기(10, 11pt)가 충분하지 않다면, `extsizes` 패키지를 이용하면 된다.

문서의 기본 글꼴 크기를 바꾸려면 위 명령어를 프리앰블에 쓴다. 이 패키지는 확장된 표준 문서 클래스 옵션을 제공하여, 8–12, 14, 17, 20 포인트 문서를 작성할 수 있도록 해준다.

### 6.1.3 글꼴 모양

이 소절에서는 글자 모양과 글자 크기, 두 가지를 다룰 것이다. 우선 글자 모양이다. 기본적인 형식은 아래와 같으며, 한글의 경우 이탤릭체 보다는 굵은 글씨를 쓰는 경우가 더 많을 것이다. 더 많은 글자 모양에 대해서는 기타 안내서를 참고하면 된다.

명령어	환경	결과
<code>\textnormal</code>	textnormal	결과
<code>\textit</code>	itshape	결과
<code>\emph</code>	없음	결과
<code>\textbf</code>	bfseries	결과
<code>\underline</code>	underline	결과

표 6.3: 글자 모양

워드프로세서에서는 폰트의 크기를 조정함으로써 글자 크기를 조정했지만, L<sup>A</sup>T<sub>E</sub>X

에서는 문서의 기본 글꼴 크기가 정해져 있고, 이에 비례하여 글자 크기를 조정한다. 글자 크기를 조정하는 명령어는 다음과 같다.

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

표 6.4: 글자 크기

#### 6.1.4 밑줄 긋기

보통 밑줄을 사용하지 않는다. 혹 사용한다 하더라도 `\underline`으로 충분하다. 그래도 꼭 다양한 형태의 밑줄을 그어야 한다면 `ulem` 패키지를 사용하면 된다. 자세한 내용은 패키지 문서를 참고하라.

## 6.2 문단 모양

### 6.2.1 들여쓰기

`\setlength{\parindent}{들여쓰기 크기}`

`\parindent` 명령을 사용하면, 한 문단의 들여쓰기 크기를 설정할 수 있다. 문단 들여쓰기를 하지 않으려면 `noindent` 명령어를 사용하면 된다.

`\usepackage[indentfirst]`

L<sup>A</sup>T<sub>E</sub>X에서는 영어 문서의 경우 장, 절의 첫번째 문단은 들여쓰기를 하지 않는 것이 기본값이다. 한글의 경우, 이것을 조정해야 하는데, 이 때 위 패키지를 사용한다. 기본값은 첫 문단 들여쓰기를 하는 것이다.

### 6.2.2 줄 간격

`\linespread{간격}`

줄 간격을 지정하는 명령어로는 `\linespread` 외에도, `setspace` 패키지를 사용하는 방법이 있다.

`\usepackage[옵션]{setspace}`

옵션으로는 `singlespace`, `onehalfspace`, `doublespace`를 지정할 수 있다.

또, `spacing` 환경을 지정하여, 줄 간격을 조정할 수도 있다. 위 두 방법이 문서 전체의 줄 간격을 지정하는 것과 달리 이 방법을 사용하면, `spacing` 환경이 적용되는 부분에서만 줄 간격이 변하고, 그 외 부분에서는 문서의 기본 줄 간격이 적용될 것이다.

`\begin{spacing}{간격}`

(본문내용)

`\end{spacing}`

### 6.2.3 문단 간격

새로운 문단을 시작하기 위해서는 다음의 두 가지 방법이 가능하다.

빈줄 넣기 : 빈 줄을 하나 이상 넣기

명령 사용 : `\par`

워드 프로세서에서 **Enter** 키를 치면 문단과 문단 사이의 공간을 만들지만, **L<sup>A</sup>T<sub>E</sub>X**에서는 문단과 문단을 나누어 줄 뿐, 문단 사이의 공간을 벌리지 않는다. 문단 간의 간격을 벌리려면, 다음 명령어를 사용한다.

`\vskip` 간격 값

그런데, 위 명령어는 문단 사이에만 효과가 있고, 첫 문단에는 효과가 없다. 이 경우 `\null`을 입력하여 문단을 띄울 위치를 잡아주어야 한다.

`\null`

`\vskip 1.3cm`

상대적인 비율로 문단 간격을 벌리려면 다음의 명령어를 사용하면 된다.

`\smallskip`

`\medskip`

`\bigskip`

문서 전체의 문단 간격을 지정하려면 다음 명령어를 사용한다.

`\setlength{\parskip}{문 단 간 격}`

### 6.2.4 정렬

오른쪽 정렬: `flushright`, `\raggedright`

왼쪽 정렬: `flushleft`, `\raggedleft`

가운데 정렬: `center`, `\centering`

다음의 예제를 따라 하고 결과를 비교해보자.

```
\begin{flushright}
2008년 2월 22일
\end{flushright}
\begin{center}
축 생일
\end{center}
\begin{flushleft}
스물 다섯번째 생일을 축하해요.
\end{flushleft}
\raggedleft 2008년 2월 22일\\
\centering 축 생일\\
\raggedright 스물 다섯번째 생일을 축하해요.\\
```

## 6.3 편집용지

### 6.3.1 문서 클래스의 옵션 활용

\documentclass의 옵션 활용

- 용지 크기: a4paper, a5paper, b5paper, letterpaper, legalpaper, executivepaper
- 용지 방향: portrait(기본값), landscape

### 6.3.2 문서여백

#### 6.3.2.1 setlength 명령어의 활용

```
\setlength[편집할 여백]{크기}
```

문서 여백을 지정하기 위해 위와 같은 명령어를 사용할 수 있다. 예제는 다음과 같다.

```

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

```

### 6.3.2.2 geometry 패키지의 활용

geometry 패키지를 사용하는 것도 편하다. 종이 크기, 여백 너비 등을 자유롭게 조절할 수 있다. 기본적인 용례는 다음과 같다.

```

\usepackage{geometry}
\geometry{papersize={25cm,35cm}}
\geometry{total={20cm,30cm}}
\geometry{body={18cm,25cm}}
\geometry{hmargin={3cm,2cm}}
\geometry{vmargin={2cm,3cm}}
\geometry{marginparwidth=2cm}
\geometry{head=1cm}

```

위 명령어에서 각 값의 순서는 표 6.5와 같다.

이를 활용하여 geometry 옵션으로 설정할 수도 있다.

```

\usepackage[left=1.0in,right=1.0in,top=1.0in,bottom=1.0in]{geometry}

```

## 6.4 머리말/꼬리말

fancyhdr 패키지를 활용하는 것이 편하다.

```

\usepackage{fancyhdr}
\pagestyle{fancy}

```

구분	1 순서	2 순서
papersize	paperwidth	paperheight
total	width	height
body	textwidth	textheight
hmargin	left	right
vmargin	top	bottom

표 6.5: 용지의 각 부분별 이름

프리앰블에 위 명령어를 넣고, 머리말과 꼬리말은 왼쪽, 가운데, 오른쪽 위치 중 원하는 곳에 입력하면 된다. 이때 명령어는 다음과 같다.

```
\lhead{}
\chead{}
\rhead{}
\lfoot{}
\cfoot{}
\rfoot{}
```

`\thepage` 명령을 입력하면 쪽 번호를 표시할 수 있다.

## 6.5 쪽 번호 매기기/새 번호로 시작

```
\setcounter{새 번호 대상}{숫자}
```

3절을 건너뛰고 4절부터 시작하도록 절 번호를 바꾸고 싶다면, 다음과 같이 쓰면 된다.

```
\setcounter{section}{3}
```

특정 값을 숫자란에 입력할 수 있다. 이때는 `\value{번호 대상}` 명령어를 활용하면 된다. 예를 들어 `exercise`라는 환경이 지정되어 있고, 이 때 번호로 쪽 번호를 사용하고 싶다면 다음과 같이 쓰면 된다.



`\setcounter{exercise}{\value{page}}`

쪽 번호의 스타일은 다음 명령어로 지정한다.

`\pagenumbering{스타일}`

스타일의 종류는 다음과 같다. arabic (1, 2, 3, ...), roman (i, ii, iii, ...), Roman (I, II, III, ...), alph (a, b, c, ...), Alph (A, B, C, ...).

## 6.6 줄 나누기/쪽 나누기

### 6.6.1 줄 나누기

`\linebreak`: 명령이 내려진 곳까지의 문장을 해당 줄의 끝까지 전개한다.

`\newline`: 명령이 내려진 곳에서 줄이 바뀐다.

`\\[줄 간격]`: 일반적으로 가장 많이 쓴다.

### 6.6.2 쪽 나누기

`\newpage`: 일반적인 쪽 나누기

`\clearpage`: 아직 처리되지 않은 모든 그림과 표를 출력한 다음 페이지를 나눈다.

`\cleardoublepage`: `\clearpage`와 같으나 다음 쪽이 항상 홀수에서 시작하도록 나눈다.

## 6.7 LyX에서의 편집

Document → Settings에서 용지와 관련한 모든 설정을 할 수 있다. Fonts에서는 기본적인 활자의 종류와 크기를 정할 수 있다. Text Layout에서는 문단간격, 줄간격을 설정할 수 있다. Page Layout에서는 용지의 종류와 가로, 세로 출력을 설정할 수 있다. Page Margins에서는 여백을 조정할 수 있다.<sup>1</sup>

---

<sup>1</sup>LyX에서는 geometry 패키지가 자동으로 적용된다. 따라서 Page Margins에서 여백을 조정하였다면 프리엠블에 geometry 패키지를 추가할 필요가 없다. 대부분의 필수적인 패키지는 LyX 환경에서 기본적

쪽나누기는 Insert → Formatting에서 New Page, Page Break 등을 선택할 수 있다.  
수식과 관련한 모든 편집은 Insert → Math에서 할 수 있다.

---

으로 구현되기 때문에, 논문을 작성하는 수준에서 프리앰블에 직접 패키지를 입력할 일은 거의 발생하지 않는다.

## 제 7 장 문서 스타일 바꾸기-명령과 클래스 작성

이 장에서는 L<sup>A</sup>T<sub>E</sub>X의 고급 기능을 배울 것인데, 이 장을 끝내고 나면 기초적인 수준에서 기존의 명령어를 수정하고, 새로운 명령어를 만들 수 있게 될 것이다. 더 나아가 자신만의 스타일로 문서를 만들 수 있도록 클래스 파일을 만드는 법도 익히게 될 것이다.

### 7.1 명령어의 정의

앞에서 명령어는 명령(command)과 환경(environment) 두 종류로 구성된다고 했다. 따라서 새로운 명령어를 만들거나, 기존의 명령어를 수정하는 명령어도 명령과 환경 각각 따로 존재한다.

```
\newcommand{명령 이름}[정수][표준값]{정의}
\renewcommand{명령 이름}[정수][표준값]{정의}
```

위 명령어를 어떻게 활용할 수 있을까? 다음의 예제를 보자.

```
\newcommand{\ex}[2][미래]{나는 #1#2에 간다.}
...
\begin{document}
\ex{대학}
\ex{출판}
\ex[] {대학}
\ex[고대]{출판}
\end{document}
```

우리는 방금 \ex라는 명령을 만들었으며, 이 명령은 정의, ‘나는 (표준값)(사용자 입력값)에 간다.’에 맞추어 표준값(‘미래’)과 사용자가 입력한 값(‘대학, 출판’), 즉 두 개의 값을 자동입력할 것이다. 위 명령어를 실행하여 결과를 확인해보자. 명령어를 새로 만드는 경우 다른 명령어를 조합하거나, 동일한 작업을 반복하여 실행할 때 편리하다.

기존의 명령을 바꾸는 경우도 생각해볼 수 있다. 다음은 한글문서를 작성할 때, 기본값으로서 장과 별도로 절 번호가 매겨지는 것(제 5 장 제 1절)을, 자신의 구미에 맞게 장 번호와 절 번호가 동시에 매겨지도록(제 5 장 5.1) 기존의 명령을 수정하는 방법이다.

```
\renewcommand\thesection{\thechapter.\arabic{section}}
```

개조식 문서의 나열 기호를 수정할 수도 있다.

```
\renewcommand*\labelitemi{\ding{43}}
```

```
...
```

```
\begin{itemize}
```

```
\item 동물
```

```
\item 식물
```

```
\item 약재
```

```
\end{itemize}
```

새로운 환경을 만들거나 기존의 환경을 바꾸는 것도 비슷하다.

```
\newenvironment{환경 이름}[정수][표준]{시작}{끝}
```

```
\renewenvironment{환경 이름}[정수][표준]{시작}{끝}
```

여기서 시작과 끝은 `\begin{환경 이름}` 이후 지정되는 시작 부분과 `\end{환경 이름}` 이후 지정되는 끝 부분의 명령어를 입력하는 곳이다.

다음의 예제는 `ex`라는 환경을 새로 만드는 것이다. 이 환경의 적용을 받는 부분은 ‘연습문제’라는 제목과 이탤릭체의 본문으로 나타날 것이다.

```
\newenvironment{exercise}{\textbf{연습 문제}\begin{itshape}}{\end{itshape}}
```

지금까지 배운 것을 기본으로, 구체적으로 문서의 스타일을 어떻게 변화시킬 수 있는지 몇 가지 예제를 보자.

### 7.1.1 대상 이름 바꾸기

이 소절에서는 한글 문서의 기본으로 설정된 ‘장, 절, 차례’ 등의 이름을 어떻게 ‘마당, 작은마당, 목차’ 등으로 바꿀 것인지 배울 것이다.

영어나 한글 문서 모두 `renewcommand`를 사용하여 ‘장, 절’의 이름을 바꿀 수 있도록 명령어를 재정의 할 수 있다. 이는 ‘표, 그림’ 등에도 적용가능하다. 아래 예제를 시험해 보라.

```
\renewcommand{\thechapter}{\Hnum{chapter}}
```

```
\renewcommand{\figurename}{Fig.}
```

한글 문서의 경우 `\usepackage{kotex}`에 한글 옵션(`[hangul]`)을 입력한 다음 `\ksnamedef`를 사용하여 이름을 바꿀 수도 있다.

```
\usepackage[hangul]{kotex}
```

• • • • •

\ksnamedef{today}{\number\year년~\number\month월~\number\day일}

\ksnamedef{contentsname}{목 ~ 차}

\ksnamedef{listtablename}{표 ~ 목 차}

\ksnamedef{listfigurename}{그림~목록}

\ksnamedef{refname}{참고 ~ 문헌}

한글 문서의 대상의 이름과 기본 표현값은 ko.T<sub>F</sub>X문서(pp. 88-9)를 참고하면 된다.

### 7.1.2 장/절 편집하기

한글 문서의 기본 값은 장, 절에 ‘제’를 붙여 ‘제 1 장, 제 1 절’로 표시하는 것이다. 이것을 영어 문서처럼 ‘1, 1’과 같이 표현하고 싶다면, 아래 명령어를 사용하면 된다. 이 경우에도 [hangu] 옵션을 사용해야 한다. 이렇게 하면 본문의 장, 절과 목차에서 ‘제’와 ‘장, 절’이 없어진다. 만약, 다른 표현으로 바꾸고 싶다면 공란으로 되어 있는 {}{}에 원하는 표현을 입력하면 된다.

```
\usepackage[hangul]{kotex}
```

• • • • •

`\kscntformat{chapter}{}{}`

\kscntformat{section}{}{}

`\ksCNTformat{subsection}{}{}`

\kscntformat{subsubsection}{}{}

또, 만약 장 표시와 장 제목을 한 줄에 넣고 싶다면, 다음의 명령어를 프리앰블에 입력하면 된다.

```
\makeatletter
\def\@makechapterhead#1{%
  \vspace*{0pt}%
  {\parindent \z@ \raggedright \normalfont
    \ifnum \c@secnumdepth >\m@ne
      \LARGE\bfseries
      \pre@chaptercnt\thechapter\post@chaptercnt
    \fi
    \quad #1%
    \vskip 40\p@
  }}
\makeatother
```

위 명령어는 복잡해 보이지만, 구조만 익히면 직접 만들 수 있다. 자세한 내용은 *A Guide to L<sup>A</sup>T<sub>E</sub>X*이나 *L<sup>A</sup>T<sub>E</sub>X Companion Book*을 참고하라.

### 7.1.3 수학 환경 지정하기

```
\newtheorem{환경 이름}{제목}[번호 지정]
```

수리적 증명을 하다보면, 증명, 정의 등 처럼 굵은 글씨와 숫자가 매겨진 항목이 필요할 수 있다. 이러한 환경을 간편하게 만들어주는 것이 `\newtheorem` 명령어이다.

{환경 이름}은 사용할 구조를 지정하며 불러내는 명령어의 역할을 하고, {제목}은 본문에서 나타나는 제목이 된다. [번호 지정]은 숫자를 매길 때 시작되는 번호를 지정하며, 옵션이다. 예를 들어 chapter를 입력하면 장 번호로 시작하게 된다. 이때 번호를 새롭게 하고 싶다면, 앞에서 사용한 `\newcounter` 명령어를 사용하면 된다.

예를 들어 다음과 같이 프리앰블에 정의해보자.

```
\newtheorem{exercise}{연습}[chapter]
```

그리고 다음과 같이 명령어를 작성하면,

```
\begin{exercise}[복습]
```

지금 까지 배운 내용을 모두 정리하여 소논문을 만들어보자.

```
\end{exercise}
```

그러면 다음과 같은 결과가 나온다.

**연습 7.1 (복습)** 지금까지 배운 내용을 모두 정리하여 소논문을 만들어보자.

또 앞에서 처럼 `\newenvironment`를 사용하여 구체적인 환경을 지정할 수도 있다. 다음 명령어는 증명의 순서를 매기고, 증명의 끝에 검은색 사각형이 나타나게 한다. 지금까지 배운 것이 어떻게 사용되고 있는지 확인해보기 바란다.

```
\newenvironment{proof}[1][증명]{\noindent\textbf{#1.}}{\rule{0.5em}{0.5em}}
```

#### 7.1.4 float 패키지의 사용

앞에서 지정한 환경을 그림이나 표처럼 만들고 싶을 때는 float패키지 환경을 사용한다. 즉 새로이 떠다니는 개체를 만든다는 것이다. 사용방법은 다음과 같다.

```
\usepackage{float}
```

```
...
```

```
\floatstyle{스타일}
```

```
\newfloat{환경 이름}{위치}{파일 확장자}[번호 지정]
```

```
\floatname{환경 이름}{본문 제목}
```

```
...
```

```
\listof{환경 이름}{목차 제목}
```

```
...
```

```
\begin{환경 이름}
```

`\floatstyle`은 스타일을 지정하는 것인데, 기본 스타일은 표 7.1에서 보는 것과 같다.

`\newfloat` 명령어의 사용은 `\newtheorem`과 같은 명령어의 사용과 동일하지만, {파일 확장자} 옵션이 추가된다. 이는 `\listof{환경 이름}{목차 제목}` 명령어를 활용하여 해당 개체에 대한 목차를 만드는 데 사용되는 파일의 확장자를 지정하는 것이다.

---

plain	그림과 표와 마찬가지로이지만, 캡션이 항상 아래에 위치한다.
boxed	상자로 처리가 되고, 캡션이 상자 아래 위치한다.
ruled	가장 상위 굵은 줄, 제목 아래 줄, 내용 끝에 줄이 그어지며, 캡션은 가장 위에 위치한다.

---

표 7.1: float 패키지의 스타일과 성격

다른 떠다니는 개체의 목차 파일처럼(표 목차 .lot; list of tables) lo를 먼저 쓰고 마지막 단어를 붙이는 것이 체계적이다.

위의 보기를 활용하여 다음을 실습해보자.

```
\usepackage{float}
...
\floatstyle{ruled}
\newfloat{exer}{!h}{loe}[chapter]
\floatname{exer}{연습문제}
...
\listof{exer}{연습문제 목차}
...
\begin{exer}
\caption{소논문 작성}
지금까지 배운 내용을 모두 정리하여 소논문을 만들어보자.
\end{exer}
```

## 7.2 클래스 파일 만들기

지금까지 배운 내용을 프리앰블에 입력해 놓아도 된다. 그러나 이 경우 프리앰블이 지나치게 길어지므로, 보기 안 좋고, 어떤 때는 프리앰블의 명령과 본문의 명령이 충돌하는 경우도 있다. 또 단어를 계산해야하는 경우 계산 오류를 일으키기도 한다. 따라서 별도의 클래스 파일을 작성하는 것이 좋다.

기존의 report 클래스를 사용하여, mythesis라는 이름의 클래스 파일을 만들어 보



자. 우선 편집기를 열고, mythesis.cls 파일을 만들자. 그 다음 아래 내용을 입력한다.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{mythesis}
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{report}}
\ProcessOptions
\LoadClass{report}
...
\endinput
```

`\NeedsTeXFormat{LaTeX2e}` 명령어로  $\text{\LaTeX}$  버전을 지정한다. 최신  $\text{\LaTeX}$ 이 설치되었다면 사용하지 않아도 된다.

`\ProvidesClass{mythesis}` 명령어로 클래스 이름을 지정하며, mythesis 대신 자신이 원하는 이름을 입력해도 된다.

`\DeclareOption*{\PassOptionsToClass{\CurrentOption}{report}}` 명령어는 옵션 변화 없이, report 클래스를 그대로 불러 온다는 뜻이다.

`\LoadClass{report}` 명령어는 report 클래스를 불러 오라는 것이다.

프리앰블에 입력했던 모든 것을 이제 이후 부분에 입력하자. 그리고 `endinput`으로 끝을 낸다. 이제 mythesis 클래스가 완성되었다. 클래스 파일을 15 쪽에서 지정한 폴더에 저장한 다음, 작업하는 문서의 클래스를 mythesis로 바꾸고 문서를 시작하면 된다.

```
\documentclass[11pt, a4paper]{mythesis}
\begin{document}
...
\end{document}
```

## 제 8 장 프리젠테이션용 파일 만들기: Beamer

L<sup>A</sup>T<sub>E</sub>X을 프리젠테이션에 사용할 수 있다. 프리젠테이션을 위해 Till Tantau에 의해 만들어진 문서 클래스가 바로 beamer class이다. 가장 간단하게는 단지 documentclass를 beamer로 지정하기만 하면 된다. 하지만 그렇게 할 경우 프리젠테이션에 그리 적당한 문서가 나오지 않는다. 따라서 beamer 클래스에서 제공하는 특수한 명령어를 통해 프리젠테이션에 적당한 문서를 만드는 것이 좋다. 자세한 내용은 beamer 가이드([http://mixing.coas.oregonstate.edu/links/latex\\_files/beamer.pdf](http://mixing.coas.oregonstate.edu/links/latex_files/beamer.pdf))를 참조하도록 하라. 이 보다 간단한 사용설명이며, 본 장 작성에 가장 많은 도움을 받은 문서는 A Beamer Tutorial in Beamer<sup>1</sup>이다.

- beamer의 장점

1. L<sup>A</sup>T<sub>E</sub>X에 익숙한 사용자라면, 동일한 명령어가 beamer에서 사용되기에 쉽게 익힐 수 있으며, L<sup>A</sup>T<sub>E</sub>X으로 작성된 글의 경우 보다 쉽게 프리젠테이션으로 사용할 수 있다.
2. 목차가 자동생성되며 각 섹션과 서브섹션으로의 링크로 연결된다.
3. 다양한 오버레이를 이용하여 다이내믹 효과를 쉽게 만들 수 있다.
4. 출력물이 pdf로 작성되기에 어떤 컴퓨터에서도 읽을 수 있어 자료 접근에 용이하다.
5. 수식을 사용하거나 한 프레임에 많은 글자가 들어 있는 경우에, Powerpoint나 Keynote에 비하여 멋진 모습을 볼 수 있다.

- beamer의 단점

1. 애니메이션이 Powerpoint나 Keynote 보다 어렵다.

---

<sup>1</sup><http://www.uncg.edu/cmp/reu/presentations/Charles%20Batts%20-%20Beamer%20Tutorial.pdf>

2. 자신이 한 작업을 보려고 한다면 매번 컴파일 해야 한다. 그것도 두번을 해야 하는 경우가 있다.

## 8.1 beamer 서식

beamer를 사용하는 가장 쉬운 방법은 이미 만들어진 서식(template)를 이용하는 것이다. 처음 beamer를 사용하는 사람에게 그리고 잠시 한 두번만 사용하려고 한다면 이 것이 최고의 해결책일 것이다. 인터넷에서 'beamer template'로 검색하면 쉽게 찾을 수 있고, 마음에 드는 것을 하나 골라서 내용만 바꾸어서 사용하면 된다. 단, 대부분 영문으로 작성된 것들이 많기에 한글을 사용한 프리젠테이션이라면 프리앰블에 `\usepackage{kotex}`를 사용하는 것을 잊지 않길 바란다.

## 8.2 프레임

beamer 클래스에서 각 장은 프레임(frame)으로 이루어진다. 프레임의 시작과 끝은 `\begin{frame}`과 `\end{frame}`으로 정해진다.

프레임에 [plain] 옵션을 사용하면 headline, footlines, sidebars 등을 사라지게 만들 수 있다. 커다란 그림을 볼때 유용하다. 참고로 프레임 제목을 지정하고 싶지 않다면, `\frametitle{}`를 사용하면 된다.

프레임의 구성은 아래와 같다.

- 머리말(Head line)과 꼬리말(Foot line)
- 왼쪽과 오른쪽의 사이드바(Left and right sidebars)
- 내비게이션 바(Navigation bars)
- 로고(Logo)
- 프레임 제목(Frame title)
- 배경(Background)
- 내용(Content)

### 8.2.1 제목 만들기

제목 프레임을 만들려면 `\titlepage` 명령어를 사용한다. 프리젠테이션의 표지가 될 슬라이드에서는, 프리젠테이션 제목, 저자, 소속기관, 발표일을 표시한다. 각각 해당하는 명령어는 `\title{제목}`, `\author{저자}`, `\institute{소속기관}`, `\date{날짜}`이다. 프리앰블에 이들 정보를 작성하지 않았다면, 첫 페이지에 나타나지 않는다.

```
\documentclass{beamer}
\title{제목}
\author{저자}
\institute{소속기관}
\date{날짜}
\begin{document}
\begin{frame}
\titlepage
\end{frame}
\end{document}
```

### 8.2.2 목차 만들기

목차는 일반 문서처럼 `\tableofcontents` 명령어를 쓴다.

```
\begin{frame}
\frametitle{Outline}
\tableofcontents
\end{frame}
```

명령어 `\tableofcontents`에 `[pausessections]` 옵션을 붙이면 목차가 순서대로 하나씩 나타나게 할 수 있다.

## 8.3 프레임 구조

다단편집과 블락설정에 대하여 알아보자

### 8.3.1 다단 편집

오른쪽에 그림을, 왼쪽에 내용을 나타내게 하고 싶을 경우, `columns` 환경을 사용한다. `columns` 환경 아래, `column` 환경의 수가 단의 수를 나타내며, 크기를 지정할 수 있다.

```
\begin{columns}[옵션]
  \begin{column}{크기}
  \end{column}
  \begin{column}{크기}
  \end{column}
\end{columns}
```

`columns` 환경의 옵션은 `b`, `c`, `t`가 있다. 각각 칼럼의 정렬을 아래, 가운데, 위로 하는 것을 의미한다. 다음의 예를 따라 해보자.

```
\begin{columns}[t]
\begin{column}{5cm}
첫 번째 칼럼
\end{column}
\begin{column}{5cm}
두 번째 칼럼
\end{column}
\end{columns}
```

`column`의 크기는 센티미터 이외에 `\column{.xx\textwidth}`를 사용하면 `xx%`로 비율을 조정할 수 있다. 아래는 각 단을 3대 7로 나누어 본 예이다.

```
\begin{columns}
\column{.3\textwidth}
첫 번째 칼럼
\column{.7\textwidth}
```

두 번째 칼럼

```
\end{columns}
```

### 8.3.2 블록 설정

프레임 안에서 block 환경을 이용하여 특정 텍스트나 그림을 상자처럼 설정할 수 있다.

```
\begin{block}{블록 타이틀}
```

내용

```
\end{block}
```

block 이외에 theorem, lemma, proof, corollary, example, alertblock, exampleblock 등의 환경도 사용할 수 있다. 설정방법은 동일하다.

## 8.4 절 편성

프리젠테이션의 절 편성은 일반문서와 동일한 명령어를 사용한다. 아래 예제를 따라해보자.

```
\begin{frame}
```

```
\section{Beamer 입문}
```

```
\subsection{Beamer란?}
```

```
\subsubsection{Beamer의 역사}
```

```
\end{frame}
```

`\subsection*{절 이름}`으로 쓰면 네비게이션 바에만 들어가고 목차에는 들어가지 않는다.

## 8.5 글꼴 모양

일반 문서에서 사용하는 글꼴모양의 명령어는 거의 동일하며, beamer에서만 구현되는 것도 있다. 다음을 연습해보기 바란다.

`\emph{연습}`, `\textbf{연습}`, `\textit{연습}`, `\textsl{연습}`, `\alert{연습}`, `\textrm{연습}`, `\textsf{연습}`, `\color{green}연습`

### 8.5.1 프로그램 코드나 공식 입력

프로그램 코드나 공식 등의 입력도 일반문서와 동일하게 `\verb + identifier character + 내용 + identifier character`의 형식으로 `\verb!내용!` 명령어를 사용하면 된다. 환경을 설정하는 경우도 동일하게 `verbatim`이다. 단, 프레임 환경에 `[fragile]` 옵션을 추가해야만 잘 작동할 수 있다는 점에 주의해야 한다. 특히 `\end{frame}`를 `verbatim` 환경내에 포함시키고자 하는 경우에는 `[fragile]` 옵션을 사용하더라도 제대로 보여줄 수 없고 `[containsverbatim]`를 사용해야 한다.

### 8.5.2 글자크기와 글꼴 테마

글자크기 변경은 일반 문서와 동일하다. 기본값으로 11pt로 설정되어 있으며, 이를 변경시키고자 한다면, `\documentclass[10pt]{beamer}`와 같이 사용하면 된다. 추가 옵션을 이용할 수 있지만, 추가 패키지를 설치하여야 하며 자세한 것은 the Beamer User Guide를 참고하길 바란다.

글꼴 테마(Font Themes)는 프리젠테이션에서 사용하는 글꼴의 속성을 변경한다. 제대로 이용하기 위해서는 각 글꼴 테마는 자신만의 옵션을 가지고 있기에 Beamer User Guide에서 옵션을 보길 바란다. `\usefonttheme{serif}` 명령어를 쓰면 된다. 다음의 글꼴 테마가 있다.

---

serif	structurebold
structureitalicserif	structuresmallcapsserif

---

표 8.1: beamer의 글꼴 테마

### 8.5.3 글꼴 묶음

자신의 프리젠테이션을 차별적으로 만들기 위해 다른 글꼴 묶음(Font Families)을 사용할수 있다. 다른 글꼴묶음을 사용하기 위해서는 우선 프리엠블에 `\usepackage{글꼴 묶음 이름}`를 추가해야 한다. 모든 글꼴 묶음이 Beamer를 설치할 때 자동 설치되지 않는다. 그러나 적어도 다음의 글꼴 묶음은 사용할 수 있다.

## 8.6 프레임 정렬, 줄간격, 띄워쓰기

프레임 정렬, 띄워쓰기, 줄간격은 L<sup>A</sup>T<sub>E</sub>X에서와 동일한 명령어가 사용된다.

### 8.6.1 프레임 상하정렬

상하정렬은 디폴트값으로 중앙([c])이 설정되어 있다. 만약, 변경하고자 한다면 윗정렬은 [t], 아래정렬은 [b]이다. 아래는 윗정렬로 변경한 예이다.

```
\begin{frame}[t]
\frametitle{Beamer}
Beamer란
\end{frame}
```

### 8.6.2 프레임 좌우 정렬

프레임의 왼쪽 정렬, 가운데 정렬, 오른쪽 정렬을 각각 `flushleft`, `center`, `flushright` 환경을 사용한다. 아래는 가운데 정렬의 예이다.

```
\begin{center}
```

serif	avant	bookman	chancery	charter	euler	helvet
mathtime	mathptm	mathptmx	newcent	palatino	pifont	utopia

표 8.2: 기본 글꼴 묶음 종류



텍스트 내용

```
\end{center}
```

### 8.6.3 줄간격과 띄워쓰기

줄간격은 명령어 `\vskip<숫자>pt`를 사용한다. 예를 들어, `\vskip15pt`는 15 포인트 수직간격을 만들어 준다. 수평간격은 `\hspace<number>pt`를 사용하면 된다. 수평간격은 문장 첫글자 띄우기와 그래픽등에 유용하다. 다른 수치 등도 사용가능하다. 센티미터는 `\vskip2cm`와 같이 사용하면 되고, 마이너스값 역시 사용하다. `\vskip-10pt` 또는 `\hspace-1cm`

## 8.7 개조식

beamer에서 개조식도 일반문서처럼 `\itemize`, `\enumerate`, `\description` 환경을 사용한 세가지 형태로 표현할 수 있다. `itemize`는 특정한 순서가 없을 때, `enumerate`는 순차적으로 숫자를 나타내는 경우에, 마지막으로 `description`은 지정된 방식으로 나타내고자 할때 사용한다.

### 8.7.1 itemize

각 새로운 아이템은 명령어 `\item`를 사용한다. 아래와 같이 사용한다.

```
\begin{itemize}
\item The first item
\item The second item
\item The third item
\end{itemize}
```

`itemize` 환경은 프레임 내에서 시작해서 끝나야만 한다. 따라서 부득이하게 하위 `item`을 이어서 해야 할 경우, `\item[]`을 이용하여 하위 아이템으로 시작한 듯 보이게 해야 한다.

```
\begin{frame}
```

```

\begin{itemize}
\item 1차 item
\begin{itemize}
\item 2-1 item
\item 2-2 item
\end{itemize}
\end{itemize}
\end{frame}

```

```

\begin{frame}
\begin{itemize}
\item[]   %%% 이 부분은 문서상에서 보이지 않는다.
\begin{itemize}
\item 2-3 item
\end{itemize}
\end{itemize}
\end{frame}

```

### 8.7.2 Enumerate

목록을 순차적인 숫자로 표현하고자 할 때 사용하게 된다.

```

\begin{enumerate}
\item The first item
\item The second item
\item The third item
\item The fourth item
\end{enumerate}

```

### 8.7.3 Description

각 아이탬은 옵션으로 기입된 문자로 표현된다.

```

\begin{frame}
\begin{description}[두 번째 아이템]
\item[첫 번째 아이템] 첫 번째 아이템 설명
\item[둘 째 아이템] 아 이 템2 설명
\item[세 번째 아이템] 세 번째 아이템 설명
\end{description}
\end{frame}

```

가입한 문자의 끝에 맞추어 정렬하려고 할 경우, 옵션을 사용할 수 있다. 기준이 되는 아이템은 정해진 명령어를 따른다 한글의 경우 첫번째에 맞추어 정렬하려면, [첫 번째 아이템]으로, 두번째 것으로 정렬하려면, [두 번째 아이템] 등으로 설정하면 된다.

## 8.8 오버레이

슬라이드 쇼의 기본값은 각 슬라이드가 한번에 모두 나타나게 하는 것이다. 그러나 마우스 클릭을 할 때마다 항목이 순차적으로 나타나게 하는 것이 효과적일 때가 많다. 실제로는 클릭 수 만큼의 문서를 생성하여 마치 클릭 할 때마다 항목이 나타나는 것처럼 보이게 처리한다. 이를 오버레이(overlay)라고 한다.

가장 단순한 것으로는 명령어 `\pause`가 있다. 보여주고 싶은 곳 바로 다음에 이 명령어를 입력하면, 프리젠테이션 시, 한 번 클릭을 할 때 그곳까지 보여준다.

물론 이전에 나타난 화면은 계속 나타나 있다. 만약에 이전에 나타난 것을 사라지게 하거나, 나타나는 순서를 바꾸고자 한다면, 명령어 `<#-#>`를 사용하면 된다. 이는 첫 번째 #에서 두 번째 #까지 나타나게 함을 뜻한다. 이 명령어 중에서 자주 사용하게 되는 것은 `<#->`로, # 순서대로 화면에 나타나게 된다. 그리고 -를 빼고 `<#>`를 사용하면 해당 #에서만 나타나게 할 수도 있다.

오버레이를 사용한 간단한 예는 아래와 같다.

```

\begin{frame}
\frametitle{오버레이}

```

---

<code>&lt; 2 - 4 &gt;</code>	2에서 4번째 클릭까지 화면에 보임
<code>&lt; -3 &gt;</code>	1에서 3까지 화면에서 보임
<code>&lt; 2- &gt;</code>	2에서 부터 끝까지 화면에 보임
<code>&lt; 2 - 4, 6, 8- &gt;</code>	2에서 4까지 나타나다가 6에서 나타났다가 사라지고, 8부터 끝까지 보임

---

표 8.3: 오버레이 사용의 기본 예

```

\framesubtitle{오버레이 실습}
\begin{block} {}
\begin{enumerate}
\item<1-> 첫 번째 줄 내용
\item<2-> 두 번째 줄 내용
\item<2> 세 번째 줄 내용
\item<1-> 네 번째 줄 내용
\end{enumerate}
\end{block}
\begin{itemize}
\item<3->{다섯 번째 줄 내용}
\end{itemize}
\end{frame}

```

가장 자주쓰게 되는 것으로 각 아이টে을 순차적으로 나타나게 하려면 [`<+->`] 를 옵션으로 주면 된다(즉, `\begin{itemize}[<+->]`). 예는 다음에 바로 다루게 된다.

### 8.8.1 오버레이와 명령어

`\item<#-#>`를 확장하여 설명할 수 있다. 각 명령어의 설명은 다음과 같다.

오버레이 응용 예는 다음과 같다.

```

\begin{frame}

```

---

<code>\item&lt;#&gt;{텍스트}</code>	#에서만 텍스트가 보임
<code>\uncover&lt;#&gt;{텍스트}</code>	#에서만 보이고, 이외에는 투명한 것 처럼 보임
<code>\only&lt;#&gt;{텍스트}</code>	#에서만 보이고, 숨겨져 있을 때, 공간을 차지하지 않음
<code>\visible&lt;#&gt;{텍스트}</code>	#에서만 보이고, 숨겨져 있을 때도 공간을 차지함
<code>\invisible&lt;#&gt;{텍스트}</code>	위 명령어의 반대
<code>\alert&lt;#&gt;{텍스트}</code>	#에서만 텍스트가 빨강색으로 강조됨
<code>\textbf&lt;#&gt;{텍스트}</code>	#에서만 텍스트가 굵게 강조됨
<code>\color&lt;#&gt;{green}텍스트</code>	#에서 텍스트가 green으로 보임
<code>\textit&lt;#&gt;{텍스트}</code>	#에서 텍스트가 기울어진 형태로 보임

---

표 8.4: 오버레이의 응용

```

\frametitle{오버레이}
\framesubtitle{오버레이 실습}
\begin{block} {}
\begin{enumerate}
\item <+-| alert@+ > 첫 번째 줄 내용
\item <+-| alert@+ > 두 번째 줄 내용
\item <+-| alert@+ > 세 번째 줄 내용
\item <+-> \color{green}네 번째 줄 내용
\end{enumerate}
\end{block}
\uncover<3->{다섯 번째 줄 내용}
\end{frame}

```

## 8.8.2 오버레이 환경

`theorem`이나 `proof` 등 대부분의 환경에서 옵션으로 `overlay`를 사용할 수 있다.

```

\begin{frame}

```

```

\begin{theorem}<1->
There exists an infinite set.
\end{theorem}
\begin{proof}<2->
This follows from the axiom of infinity.
\end{proof}
\end{frame}

```

## 8.9 표와 그림

### 8.9.1 표 만들기

간단한 표를 `tabular` 환경에서 만들 수 있다. 기본적인 양식은 일반 문서에서의 표 작성과 같다.

```

\begin{tabular}{|c|cc|}
\hline
cell 1 & cell 2 & cell 3 \\
\hline \hline
cell 4 & cell 5 & cell 6 \\
\hline
\end{tabular}

```

### 8.9.2 표를 오버레이 시키기

우선 프리앰블에 `\usepackage{colortbl}`를 추가한 후에, 열을 한줄씩 보이기 위하여는 `\pause`, 행을 한줄씩 보이기위해서는 `\onslide<n->` 명령어를 사용하면 된다. 예는 아래와 같다.

```

\begin{tabular}{lcccc}
& A & B & C & D \\ \hline
X & 1 & 2 & 3 & 4 \pause \\
\end{tabular}

```

```

Y      & 3 & 4 & 5 & 6 \pause\\
Z      &5&6&7&8
\end{tabular} \label{rv:overlaytablerow}

\begin{tabular}{lc<{\onslide<2->}c<{\onslide<3->}c<{\onslide<4->}c<{\onslide}c}
Class & A & B & C & D \\
X      & 1 & 2 & 3 & 4 \\
Y      & 3 & 4 & 5 & 6 \\
Z      &5&6&7&8
\end{tabular}{rv:overlaytablecolumn}

```

### 8.9.3 그림 삽입

일반문서와 동일한 방식으로 외부에서 그림 불러오기를 할 수 있다. .pdf, .jpg, .jpeg, .png 형식의 파일을 지원하므로 다른 형식은 해당 형식으로 변환후에 사용해야 한다. 프리앰블에 `\usepackage{graphics}`를 반드시 기입해야 한다는 점은 잊지않도록 한다. 명령어는 당연히 `\includegraphics`를 사용한다.

## 8.10 테마

테마를 변경하면 프리젠테이션의 전체적인 모양과 느낌이 완전히 변화한다.

### 8.10.1 전체 테마

Antibes, Boadilla, Frankfurt, Juanlespins, Montpellier, Singapore, Bergen, Copenhagen, Goettingen, Madrid, Paloalto, Berkeley, Darmstadt, Hannover, Malmoe, Pittsburgh, Berlin, Dresden, Ilmenau, Marburg, Rochester, Warsaw 등 여러 테마들이 있으며, 이들은 인터넷 검색을 통하여 그 모습을 쉽게 확인할 수 있다. 예를 들어 Warsaw를 사용하고 싶다면 프리앰블에 `\usetheme{Warsaw}`를 써놓으면 된다.

### 8.10.2 색 테마

어떤 테마의 형태는 마음에 들지만 색이 싫다면, 색 테마(Color Theme)를 사용하여 변경할 수 있다. 색 테마를 사용하려면, 프리앰블에 `\usecolortheme{}`을 설정하고, albatross, crane, beetle, dove, fly, seagull, wolverine, beaver 중 하나를 선택하면 된다.

그러나 단지 프레임 안에 사용되는 블록의 색을 바꾸고 싶다면(이를 내부 색 테마(Inner Color Theme)라고 함), lily, orchid, rose 중에서 하나를 선택하면 된다(예, `\usecolortheme{lily}`)

머리말, 꼬리말, 사이드바의 색 조합(palette)을 변경하고 싶다면(이를 외부 색 테마(Outer Color Theme)라고 함), whale, seahorse, dolphin 중에 하나를 선택하면 된다(예, `\usecolortheme{whale}`).

## 8.11 유인물 출력

유인물 형태로 출력하고 싶은 경우 `\documentclass`에 옵션으로 ‘handout’을 넣어준다(`\documentclass[handout]{beamer}`). 이렇게 하면 오버레이 지정으로 인해 늘어난 장 수를 한 장에 다 포함되도록 맞추어 준다. 색의 수를 줄이고 싶다면 테마를 바꾸어 준다. 프리앰블을 다음과 같이 수정하면, 출력용지 설정과 함께 한 페이지에서 출력되는 슬라이드의 갯수를 조정할 수 있다. 아래에는 레터지 한 장에 2개의 슬라이드를 출력하게 된다.

```
\usepackage{pgfpages}
\pgfpagesuselayout{2 on 1}[letterpaper]
```

## 8.12 LyX에서 Beamer 구현

LyX의 장점을 하나 더 들자면 Beamer를 작성하기 쉽다는 점이다. Document → Settings → Document Class → presentation (beamer)로 변경하면 Beamer의 작성이 가능하다.

Beamer 테마의 적용은 역시 Document → Settings → L<sup>A</sup>T<sub>E</sub>X Preamble에서 이루어진다. 그리고 Beamer 작성에서 필요한 몇 가지 L<sup>A</sup>T<sub>E</sub>X 코드는 Insert → T<sub>E</sub>X Code로 삽입할



수 있다.<sup>2</sup>

---

<sup>2</sup>LyX에서 제공하는 T<sub>E</sub>X Code 기능은 텍 명령어를 알고 있는 사용자에게 유용하다. LyX의 세부설정을 잘 모르는 대부분의 경우에서 T<sub>E</sub>X Code의 삽입만으로 이를 해결할 수 있기 때문이다.

## 제 9 장 그래프 그리기

### 9.1 curves 패키지의 활용

이 절에서는 `curves` 패키지를 활용하여, 예산제약식과 무차별 곡선을 그리는 방법을 익힐 것이다.

우선 `curves` 패키지를 설치하고 프리앰블에 패키지 사용을 지정한다 (`\usepackage{curves}`). 그리고 다음 명령어를 입력해 보자.

```
\begin{figure}[htbp]
\setlength{\unitlength}{1mm}
\begin{center}
\begin{picture}(55,55)
\put(0,0){\vector(0,1){50}}
\put(0,51){\makebox(0,0)[b]{y}}
\put(0,0){\vector(1,0){50}}
\put(52,0){\makebox(0,0)[c]{x}}
\thicklines
\curve(5,48,20,20,48,5)
\put(10,40){무 차 별 곡 선}
\put(20,20){\circle*{1.5}}
\put(23,20){\makebox(0,0)[b]{A}}
\curve(0,40,40,0)
\put(10,8){예 산 제 약 식}
\end{picture}
\caption{curves.sty를 사용한 예산제약식과 무차별 곡선}
\label{fig:curves}
```

```
\end{center}
\end{figure}
```

각각의 명령어를 살펴보면,

`\unitlength{단위}`: 단위 간격을 지정한다. 이 예제에서는 1mm를 단위로 했다.

`\begin{picture}(가로,세로)`: 괄호안의 숫자는 그림의 전체 크기, 즉 가로, 세로 크기를 결정한다. 이 예제에서는 기본 단위간격이 1mm 이므로 5.5cm×5.5cm의 캔버스가 그려진다고 생각하면 되겠다.

`\put(위치){내용}`: 이 명령어는 문자 또는 그림을 삽입하기 위해 사용된다. 앞에서 그린 캔버스를 모눈종이라고 생각할때, 소괄호안의 숫자는 입력하고자 하는 문자의 내용이 나타날 위치를  $x, y$  좌표로서 잡아준다고 생각하면 된다. 따라서 예제의 `\put(20,20){\circle*{1.5}}` 명령어는 (20,20)에 지름 1.5의 원을 그리라는 것이다.

`\curve{좌표}`: 이 명령어는 곡선을 그릴 것이다. 곡선을 그리기 위해서는 적어도 세 개의 좌표가 필요하다. 예제에서 `\curve(5,48,20,20,48,5)`는 (5,48) 점에서 출발하여, (20,20) 점을 지나, (48,5) 점에 도달하는 곡선을 그리게 된다. 그러므로 시작점과 끝점만 지정한 `\curve(0,40,40,0)`는 직선만 표시하게 될 것이다.

예제의 결과는 다음과 같이 나타난다.

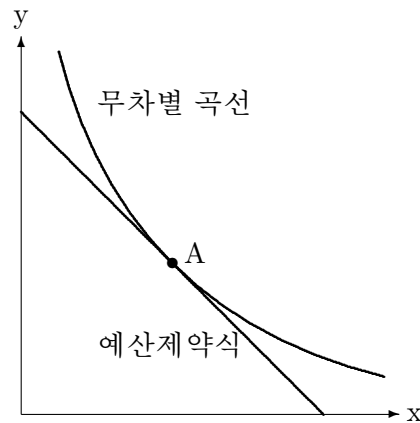


그림 9.1: curves.sty를 사용한 예산제약식과 무차별 곡선

## 9.2 PSTricks의 활용

### 9.2.1 설치 및 패키지 사용 지정

`curves` 패키지를 사용하면 간단한 그래프를 그릴 수 있지만, 복잡한 그래프를 그리는 데에는 한계가 있다. 복잡한 그래프를 그리기 위해서는 PSTricks가 가장 강력한 도구가 될 수 있다. 2012년 현재, PSTricks는 T<sub>E</sub>X Live 2011을 설치하면 기본 패키지로 같이 설치된다.

PSTricks 패키지를 사용하기 위해서는 프리앰블에 사용을 지정해주어야 한다. 다음의 패키지 이외에도 PSTricks를 활용하는 패키지는 많이 있으며, 검색을 통해 확인할 수 있다.

```
\usepackage{pstricks}
\usepackage{pst-plot}
\usepackage{pst-tree}
\usepackage{pst-eps}
\usepackage{pst-node}
\usepackage{pst-func}
\usepackage{pstricks-add}
\usepackage{multido}
```

### 9.2.2 컴파일링과 주의사항

Windows의 KC2007을 사용하는 경우, 다음 절차에 따라 컴파일링 하면 된다.

1. 기본적인 컴파일링을 한다.
2. KC2007의 오른쪽 아래에 있는 PS 관련 컴파일링에서 DVI → PS 로 컴파일링을 한다.
3. PS로 보기를 하거나, PS → PDF 로 컴파일링을 한 번 더 한다.

Windows에서 T<sub>E</sub>Xworks를 사용하는 경우 컴파일 옵션에서 latex+dvips+pstopdf 나 XeLaTeX으로 전환하여 컴파일 하면 된다.

Mac에서 TeXShop을 사용하는 경우, 다음의 두 가지 방법이 있다.

1. 한글 구현 명령어를 수정하지 않는 경우

- (a) `\documentclass[dvips]{article}` 처럼 dvips 옵션을 입력한다.
- (b) `\begin{document}` 명령어 다음 줄에 `\pdfoutput -1`를 입력한다.
- (c) 컴파일링한다.
- (d) 결과가 나타나지 않을텐데, 작업 폴더로 가면 .dvi 확장자를 가진 파일이 있다. 이것을 더블클릭하면 결과물이 나타난다.

2. 한글 구현 명령어를 수정하는 경우

- (a) 15 쪽의 한글 입력 명령어 중 `\usepackage[unicode,dvipdfm]{hyperref}`의 옵션에서 dvipdfm을 dvips로 수정한다.
- (b) 메뉴에서 Typeset → TeX and DVI를 선택한다.
- (c) 컴파일링한다.

이제 컴파일이 가능할 것이다. 하지만, PSTricks를 사용하는 경우, 문서에 삽입할 다른 그림은 반드시 확장자가 eps인 그림 파일로 변환되어 있어야 한다. 그렇지 않다면, 그림이 공백으로 보일 것이다. Photoshop을 사용하면 가능하며, 그외 어플리케이션은 검색을 통해 찾을 수 있다.

### 9.2.3 PSTricks 활용의 기초

#### 9.2.3.1 그래프 그리기

```
\begin{figure}[!hbt]
\begin{center}
\psset{yunit=0.5cm,xunit=0.5cm}
\begin{pspicture}(-1,-1)(7,6)
\psaxes[>->](6,5)
\uput[0](6,0){$x$}
\rput[0](3,3){$(3,3)$}
```

```

\uput[90](0,5){$f(x)$}
\end{pspicture}
\end{center}
\end{figure}

```

위 명령어를 하나씩 살펴보자. 먼저 그림을 정의하고, 그 다음 `[!hbt]`에서 h는 here를 의미하며 떠다니는 개체 명령이 사용된 바로 그 자리를 이른다. t는 top을, b는 bottom을, p는 page를 의미한다. !는 떠다니는 개체의 위치를 제어하는 내부변수들을 고려하지 않도록 하는 것이다.

```

\begin{figure}[!hbt]
\end{figure}

```

yunit과 xunit은 y축과 x축의 정수단위의 크기를 지시하는 것으로 x축과 y축의 눈금 하나를 0.5cm로 표현하는 것이다.

```

\psset{yunit=0.5cm,xunit=0.5cm}

```

$(-1,-1)(7,6)$ 는 그림의 x축의 크기가 -1에서 7까지, y축의 크기가 -1에서 6까지를 의미한다. 즉, 좌표를  $(-1,-1)$ 과  $(7,6)$ 이 이루는 사각형에서 표현하자는 것이다.

```

\begin{pspicture}(-1,-1)(7,6)
\end{pspicture}

```

이 표현은 원점  $(0,0)$ 과 x축의 6, y축의 5까지 화살표로 축을 그리는 것을 의미한다. `[ ]`의 옵션에 따라 여러가지로 축을 표현할 수 있다.

```

\psaxes[ ]{->}(6,5)

```

자 이제는 보통 경제학에 표현하는 x축과 y축만을 표현하고 눈금은 없는 그래프는 다음과 같이 그릴 수 있다.

```

\begin{figure}[!hbt]
\begin{center}

```

```

\psset{yunit=1cm,xunit=1cm}
\begin{pspicture}(-1,-1)(7,6)
\psaxes[labels=none,ticks=none]
    {->}(0,0)(-.5,-.5)(6,5)
\uput[0](6,0){$x$}
\psline[linecolor=red,
    linestyle=dashed](1,1)(5,4)
\uput[90](0,5){$f(x)$}
\end{pspicture}
\end{center}
\end{figure}

```

$Ox=number$ ,  $Oy=number$ 는 원점을 변경하고,  $Dx=number$ ,  $Dy=number$ 는 좌표의 크기를 선택할 수 있다. 또한, 눈금을 없애려면  $ticks=none$ , 라벨을 없애려면  $labels=none$ 를 선택하면 된다.

`uput`은 글자를 입력하기 위해 사용하는 조판명령어이다.  $[0]$ 은 동경각을 나타낸다. 따라서,  $[90]$ 은  $(6,0)$ 좌표의 위쪽에,  $[-90]$ 은 좌표의 아래쪽을 의미한다. `rput`은 그 위치에 조판하게 한다.

### 9.2.3.2 에지워스 상자 그리기

```

\begin{figure}[!hbt]
\psset{yunit=1cm,xunit=1cm}
\begin{center}
\begin{pspicture}(-2,-1)(6,3)
\psgrid[style=gridstyle,gridlabels=5pt]
\psaxes[ticks=none]{->}(0,0)(0,0)(5,2.5)
\psaxes[ticks=none]{<-}(4,2)(-1,-.5)(4,2)
\uput{3pt}[dl](0,0){$0^A$}
\uput{3pt}[ur](4,2){$0^B$}
\uput[r](5,0){$x^A$}

```

```

\uput[u](0,2.5){$y^A$}
\uput[l](-1,2){$x^B$}
\uput[d](4,-.5){$y^B$}
\end{pspicture}
\end{center}
\end{figure}

```

### 9.2.3.3 선분 그리기

```

\psline[linecolor=red,linestyle=dashed](1,1)(5,4)

```

(1,1)에서 (5,4)까지 빨간색 점선으로 선분을 그리는 것이다.

### 9.2.3.4 그래프 2개 그리기

그래프 2개를 그리는 것은 minipage를 이용하여 조판할 수 있다. 아래의 명령어로 두 개의 그래프를 조판할 수 있다.

```

\begin{figure}[!hbt]
\psset{yunit=0.7cm,xunit=0.7cm}
\begin{center}
\begin{minipage}{.45\textwidth}
\begin{pspicture}(-1,-1)(7,6)
\psaxes[labels=none,ticks=none]{->}(0,0)(-.5,-.5)(6,5)
\uput[0](6,0){$x$}
\uput[90](0,5){$f(x)$}
\uput[0](4,1){(1)}
\psline[linecolor=red,linestyle=dashed](1,4)(4,1)
\end{pspicture}
\end{minipage}
\hfill
\begin{minipage}{.45\textwidth}

```



```

\begin{pspicture}(-1,-1)(7,6)
\psaxes[labels=none, ticks=none]{->}(0,0)(-.5,-.5)(6,5)
\uput[0](6,0){$x$}
\uput[90](0,5){$f(x)$}
\uput[0](4,4){(1)}
\psline[linecolor=blue](1,1)(4,4)
\end{pspicture}
\end{minipage}
\end{center}
\caption{그래프 2개 그리기} \label{pic:pic3}
\end{figure}

```

#### 9.2.3.5 함수를 이용한 그래프

psstriks-add를 이용하여 함수 표현을 쉽게 할 수 있다. algebraic을 이용하여  $1.5 * (x - 3)^2 + 2$ 를  $1.5 \leq x \leq 4.5$ 에서 그릴 수 있다.

```

\begin{center}
\psset{yunit=1cm,xunit=1cm}
\begin{pspicture}(-1,-1)(7,6)
\psaxes[labels=none, ticks=none]{->}(0,0)(-.5,-.5)(6,5)
\psplot[linecolor=blue, linewidth=1.2pt, algebraic]{1.5}{4.5}{1.5*(x-3)^2 +2}
\uput[0](6,0){$x$}
\uput[90](0,5){$f(x)$}
\psline[linecolor=red, linestyle=dashed](1.5, 2)(4.5,2)
\uput[-90](5,2){$f'(x_0)=0$}
\uput[-90](5,1){$f''(x_0)>0$}
\psline[linestyle=dashed](3,2)(3,0)
\uput[-90](3,0){$x_0$}
\end{pspicture}
\end{center}

```

### 9.2.3.6 음함수 그리기; psplotImp

다음은 음함수를 그리기 위한 명령어이다. 다음 명령어는  $(x - 1.84)^2 + (y - 3.23)^2 = 1$ 인 원을 그리는 것이다. (0.83, 2.22)(2.85, 4.25) 영역 내에서 음함수의 그래프를 표현하는 것이다.

```
\begin{figure}[!hbt]
\psset{yunit=1cm,xunit=1cm}
\begin{center}
\begin{pspicture}(-1,-1)(7,6)
\psaxes[Dy=10,Dx=10]{->}(0,0)(-.5,-.5)(6,5)
\psplot[linecolor=blue,linewidth=1.2pt,algebraic]
{0.8}{5.5}{(4/10)*(x-1)*(x-3)*(x-5)+2}
\uput[0](6,0){$x$}
\uput[90](0,5){$f(x)$}
\psplotImp[linecolor=red,linewidth=0.5pt,algebraic]
(0.83,2.22)(2.85,4.25){(x-1.84)^2+(y-3.23)^2-1}
\psplotImp[linecolor=green,linewidth=0.5pt,algebraic]
(3.13,-0.25)(5.16,1.76){(x-4.15)^2+(y-0.75)^2-1}
\end{pspicture}
\end{center}
\end{figure}
```

### 9.2.3.7 무차별곡선과 예산제약 그래프

이제, 무차별 곡선과 예산제약선을 그려보자.

```
\begin{figure}[!hbt]
\psset{yunit=1cm,xunit=1cm}
\begin{center}
\begin{pspicture}(5,4)
\psgrid[style=gridstyle]
```

```

\psset{algebraic=true}
\multido{\Iu=1+1}{4}{%
\psplotImp(0,0)(5,4){sqrt(x*y)-sqrt(\Iu)}}
\psplotImp(0,0)(5,4){3*x+4*y-12}
\psdots(2,1.5)
\psaxes[ticks=none]{->}(5,4)
\end{pspicture}
\end{center}
\end{figure}

```

## 9.3 LaTeXDraw의 활용

### 9.3.1 LaTeXDraw

PSTricks는 강력한 그래픽 도구이지만, 종이를 모눈종이로 보고 좌표를 지정해주며 그림을 그리는 방식이어서, 많은 시행착오를 겪어야 원하는 그림을 그릴 수 있다. 또 다른 패키지와의 자주 충돌하는 편이다. 따라서 LaTeXDraw를 설치한 후, LaTeXDraw에서 작성한 그림을 pdf로 변환저장한 후  $\text{T}_\text{E}\text{X}$ 문서에 삽입하기를 권한다.

### 9.3.2 설치

LaTeXDraw는 JAVA로 작성된 멀티플랫폼 애플리케이션이며, 오픈소스로 개발중이다. <http://latexdraw.sourceforge.net/download.html>에서 다운받을 수 있다. 압축파일을 다운 받아 푼 후, install.jar를 실행시키면 된다. 경우에 따라서는 JAVA 애플릿이나 X11을 설치해야할 필요가 있다.

설치가 완료된 후, LaTeXDraw를 실행하면, 모눈종이가 그려진 그림판이 나타나게 되고, 그림판 위에 그래픽 도구들이 있다. 이를 활용하여, 그림을 그리면 된다. 오른쪽의 창을 보면 그림이 그려짐에 따라, 명령어가 생성되는 것을 알 수 있다. LaTeXDraw는 눈으로 보고 있는 그래프를 PSTrick 명령어의 그림으로 자동으로 바꾸어 주는 역할을 한다.

### 9.3.3 eps 파일 생성

앞으로 설명하려는 것은 LaTeXDraw로 생성한 명령어를 eps 파일로 전환시키는 과정이다. 전체과정은 다음과 같다.

1. tex 파일을 컴파일 하여, EPS 파일을 생성한다.
2. 생성된 EPS 파일을 사용하고자 하는 tex 파일에 그림으로 삽입한다.

먼저 tex 파일을 만들자. 파일이름은 일단 latexdraw로 하자.<sup>1</sup> 이 파일에 다음의 내용을 집어넣자.

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

```
\usepackage{pstricks}
```

```
\usepackage{pst-plot}
```

```
\usepackage{pstcol}
```

```
\usepackage{pst-node}
```

```
\usepackage{pstricks-add}
```

```
\usepackage{pst-grad}
```

```
\usepackage{pst-func}
```

```
\usepackage{pst-tree}
```

```
\usepackage{pst-text}
```

```
\usepackage{pst-char}
```

```
\usepackage{pst-eps}
```

```
\begin{document}
```

```
\begin{TeXtoEPS}
```

---

<sup>1</sup>어떤 파일이름을 써도 좋지만, pstricks는 사용할 수 없다. pstricks.tex을 pstricks 패키지에서 먼저 읽어들이기 때문에, 컴파일할 때 two documentclass 라는 에러가 발생한다.

```
\end{TeXtoEPS}
```

```
\end{document}
```

pst로 시작하는 것은 pstricks 관련 패키지를 의미하며, 필요에 따라 늘려 나가도 된다. eps 파일로 전환하는 과정에서 가장 중요한 것은 \usepackage{pst-eps} 이다. 다음 본문의 \begin{TeXtoEPS}와 \end{TeXtoEPS} 사이에, 그림으로 전환하고자 하는 명령어를 LaTeXDraw의 창에서 복사하여 집어넣자.

이제 dvi 파일을 추출해야 한다. Windows의 TeXWorks를 사용하는 경우, TeXworks를 실행한 후, ‘edit → preferences → typesetting → Processing tools 추가 → pdfLaTeX w/ -shell-escape 으로 명령 → miktex-pdfTeX.exe 으로 프로그램 선택 → arguments 에 다음 순서대로 입력 -shell-escape, -synctex=1, -undump=pdfLatex, \$fullname’ 한 후, ‘pdfLaTeX w/ -shell-escape’으로 컴파일 하면 된다. Mac의 TeXShop을 쓰는 경우, TeX and DVI을 선택하여 컴파일하면 된다.

작업폴더를 보면, latexdraw.dvi 파일이 생성된 것을 확인할 수 있다. 이 dvi 파일을 eps 파일로 바꿔주어야 한다. latexdraw.pdf 파일도 같이 생성되지만, 이 파일은 인쇄 용지 전체에 그림을 그린 형태로 나타나므로, 그림만 뽑아내기 위해서는 다음 과정을 더 거쳐야 한다.

Windows의 경우, ‘시작 → 실행 → cmd’ 를 입력하여, 명령창을 띄우고, Mac의 경우, ‘Application → Utilities → Terminal’ 을 실행시킨다. 현재 작업 중인 폴더로 변경한다.<sup>2</sup>

이제 명령창에 다음의 명령어를 입력한다.

```
dvips -P outline -GO -E -o latexdraw.eps latexdraw.dvi
```

dvips는 dvi 파일을 eps 파일로 바꾸어 주라는 명령어이고, 바뀌진 파일의 이름과 바꾸어야 할 파일의 이름을 마지막에 적어준다. 이제 생성된 eps 파일을 4에서 배운 방법대로 작업 중인 파일의 원하는 위치에 삽입하면 된다.

---

<sup>2</sup>OS X의 경우, System preferences → Keyboard → Keyboard Shortcuts → Services → Files and Folders → New Terminal at Folder를 체크해준다. Finder에서 작업폴더를 선택하고, 마우스 우클릭 (option + click 또는 두 손가락을 대고 클릭) 을 하면 목록에 Services가 보이고, New Terminal at Folder를 선택하면, 현재 작업폴더에서 터미널을 실행할 수 있다.

## 9.4 그외의 사항들

추가 사항은 <http://tug.org/PSTricks/main.cgi?file=examples>을 참조하기 바란다. 또한 1장에서 소개한 (1) PSTricks를 이용한 함수의 플로팅: 효용극대화모형을 중심으로, (2) PostScript Drawing: An Economist's Guide, (3) PSTricks: PostScript macros for Generic TeX을 참고하기 바란다.

## 제 10 장 경제학을 위한 L<sup>A</sup>T<sub>E</sub>X

이 장에서는 경제학 논문을 작성하며 자주 사용하게 되는 L<sup>A</sup>T<sub>E</sub>X 팁을 다룬다. 패키지를 이용한 게임 보수표 작성, Stata의 결과값 불러오기, 참고문헌 손쉽게 달기를 기술한다.

### 10.1 게임 보수표 및 게임 나무의 작성

4.2 절에서 표를 이용하여 게임 보수표를 작성하는 것을 익혔다. 이 절에서는 `sgamevar`와 `egameps` 패키지를 사용하여 좀 더 쉽게 게임 보수표와 게임 나무를 작성하는 법을 익힐 것이다.

우선 두 패키지를 <http://www.economics.utoronto.ca/osborne/latex/index.html><sup>1</sup>에서 다운 받은 후, 15쪽에서 지정한 위치에 파일을 복사한다.

그리고, 언제나처럼 프리앰블에 패키지의 사용을 지정한다.

```
\usepackage{sgamevar, egameps}
```

자세한 사용방법은 위 홈페이지에서 관련 매뉴얼을 다운받아 보면 된다. 여기서는 가장 기초적인 용례만 설명할 것이다.

#### 10.1.1 전략형 게임 보수표의 작성

전략형 게임을 그리는 것을 표를 작성하는 것과 비슷하다. 다만 셀의 분리를 지정하는 명령어로 `\>`을 사용한다. 표와 마찬가지로 `&` 명령어를 사용하고 싶다면, `sgame.sty` 패키지를 다운 받으면 된다. 이 패키지도 위에서 소개한 홈페이지에 있다. `sgamevar.sty`를 다운 받으면 이 파일을 열고<sup>2</sup>, 가장 마지막 두 줄이 빈줄이라면 이를 삭제한다. 이 부분은 인코딩 충돌을 일으킬 것이다.

가장 중요한 부분은 다음 명령어이다.

---

<sup>1</sup>본문을 잘 살펴보면 패키지를 다운 받을 수 있도록 하이라이팅이 되어 있다.

<sup>2</sup>아무 텍스트 편집기에서나 열릴 것이나, 현재 사용중인 T<sub>E</sub>X용 편집기를 사용할 것을 권한다.

```
\begin{game}{열 개수}{행 개수}[행위 자][행위 자]
...
\end{game}
```

게임을 작성하는 용례는 다음과 같다.

```
\begin{table}[!h]
\begin{center}
\begin{game}{2}{2}[Player 1][Player 2]
\> $L$ \> $M$\\
$T$ \> $2,2$ \> $2,0$\\
$B$ \> $3,0$ \> $0,9$
\end{game}
\caption{전략형 게임 보수표}
\end{center}
\end{table}
```

결과는 다음과 같이 나온다.

		Player 2	
		<i>L</i>	<i>M</i>
Player 1	<i>T</i>	2, 2	2, 0
	<i>B</i>	3, 0	0, 9

표 10.1: 전략형 게임 보수표

### 10.1.2 확장형 게임 나무의 작성

확장형 게임은 9 장에서 배운 PSTricks를 활용하여 그린다. 따라서, 기본적인 명령어 구성은 PSTricks와 유사하다. 하지만, PSTricks 패키지를 프리앰블에 정의할 필요는 없다.



확장형 게임은 전개형 게임에 비해 다소 복잡하고, 일일이 그림을 확인해야 할 필요가 있다. 이 소절에서 소개하는 기본적인 용례 외에 다양한 형태의 게임 나무는 위에서 소개한 홈페이지의 매뉴얼과 예제 파일을 확인하면 된다. 기본 명령어는 다음과 같다.

```
\begin{figure}[htb]
\hspace*{\fill}
\begin{egame}(600,280)
\putbranch(300,240)(2,1){200}
\iib{1}{\mathbb{L}}{\mathbb{R}}
\putbranch(100,140)(1,1){100}
\iib{}{\mathbb{a}}{\mathbb{b}}[\$1,0\$][\$2,3\$]
\putbranch(500,140)(1,1){100}
\iib{}{\mathbb{c}}{\mathbb{d}}[\$0,1\$][\$-1,0\$]
\infoset(100,140){400}{2}
\end{egame}
\hspace*{\fill}
\caption{확장형 게임}\label{fig:extengame}
\end{figure}
```

다음 명령어가 전체적인 게임의 크기를 지정한다.

```
\begin{figure}[위 치]
\begin{egame}(전체 크기 지정)
\end{egame}
\end{figure}
```

보기 예에서 `\begin{egame}(600,280)`은 (600,280)이라는 공간을 확장형 게임에 할당하겠다는 것이다.

`putbranch` 명령은 앞에서 지정한 전체 그림에서 어느 위치에서 시작할지, 그리고  $x, y$  축으로 표시된 방향, 수평으로 뻗는 선의 길이를 지정한다.

```
\putbranch(시작위치)(x,y 방향){수평 길이}
```

보기 예에서 `\putbranch(300,240)(2,1){200}`은 (300,240)의 위치에서, x, y축으로 각각 (2,1) 방향으로 선을 그리는 데 그 길이는 200이라고 지정하는 것이다.

`iib` 명령은 행위자, 행위, 보수값을 지정하는 데 사용된다.

`\iib{행위자}{행위}{행위}[보수값][보수값]`

보기 예에서 `\iib{}{$a}{$b}[$1,0][$2,3]`은 행위자 1은 표시에서 제거하고, 행위 a, b를 선택했을 때, 각각 보수값이 (1,0), (2,3)이 된다는 것이다.

정보집합은 `infoset` 명령어를 사용하여 지정한다. 명령어는 다음과 같다.

`\infoset(시작위치){수평 길이}{행위자}`

보기 예에서 `\infoset(100,140){400}{2}`은 (100, 140)에서 출발하여, (500,140)까지 정보 집합을 그린다는 것이다.

용례의 결과는 다음과 같이 나타난다.

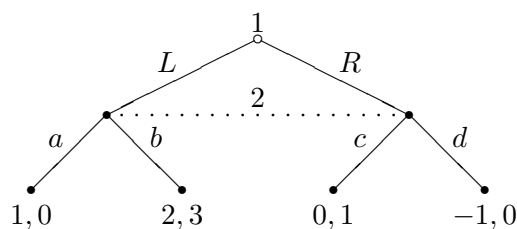


그림 10.1: 확장형 게임

## 10.2 Stata 결과값 불러오기

### 10.2.1 sutex, outtex 명령어

Stata의 결과값을 불러오는 명령어는 표 10.2.1와 같다. 모든 명령어는 ado 파일의 추가 설치가 필요하다. stata에서 ado 파일의 추가하려면, stata의 콘솔창에서 `findit ado` **파일 이름**을 입력한 후 설치 관련 링크를 클릭하면 된다.

이 절에서는 `sutex`, `outtex`의 기본적인 용례만 소개한다. 더 자세한 옵션과 기타 명령어는 Stata 홈페이지(<http://www.ats.ucla.edu/stat/stata/latex/default.htm>)를 참고하면 된다.

명령어	기능
<code>listtex</code>	<code>list</code> 명령어의 결과를 L <sup>A</sup> T <sub>E</sub> X 언어로 변환
<code>latabstat</code>	<code>tabstat</code> 명령어의 결과를 L <sup>A</sup> T <sub>E</sub> X 언어로 변환
<code>sutex</code>	<code>summarize</code> 명령어의 결과를 L <sup>A</sup> T <sub>E</sub> X 언어로 변환
<code>graph2tex</code>	그래프를 eps 파일로 추출, 관련 L <sup>A</sup> T <sub>E</sub> X 언어로 변환
<code>outtable</code>	결과값 등을 L <sup>A</sup> T <sub>E</sub> X용 행렬식으로 전환

표 10.2: Stata 결과의 L<sup>A</sup>T<sub>E</sub>X전환 명령어

`sutex` 명령어를 다음의 절차에 따라 사용해보자.

```
use http://www.ats.ucla.edu/stat/stata/notes/hsb2, clear
sutex
sutex , minmax
```

`sutex` 명령어를 실행한 다음 나온 L<sup>A</sup>T<sub>E</sub>X용 코드를 문서에 삽입한 후 컴파일 하면 된다. `minmax` 옵션을 사용하면 최대, 최소값도 같이 보여준다. 표의 이름, 상호참조, 형태 등은 자신의 기호에 맞게 또는 저널의 양식에 맞게 수정하면 된다.

`outtex` 명령어도 이와 유사한 절차로 사용할 수 있다.

```
use http://www.ats.ucla.edu/stat/stata/notes/hsb2, clear
regress write math female
outtex
outtex, level
```

`level` 옵션을 사용하면 유의수준도 같이 표시한다. 유의수준을 표기하는 방식을 바꾸고 싶다면, `\def` 이하를 수정하면 된다.

### 10.2.2 Stata 응용 및 기타 통계 패키지

`texdoc`은 ado 파일로 Stata 9.2이후 버전에 설치하여 사용할 수 있다. 설치하는 Stata 콘솔창에서 `findit texdoc`을 입력하라. 이를 사용하여 Stata의 콘솔창에서 바로 L<sup>A</sup>T<sub>E</sub>X

문서를 작성할 수 있다. 자세한 내용은 다음 문서를 참고하라(<http://fmwww.bc.edu/repec/dsug2009/jann.pdf>).

R, SAS 등 다른 통계패키지도 Stata에 내장된 명령어와 유사한 기능을 지원하며, 이는 검색을 통해 확인 바란다.

R은 Sweave를 이용하여, L<sup>A</sup>T<sub>E</sub>X 안에서 문서 편집 및 통계 처리를 동시에 할 수 있도록 지원한다. R의 결과를 L<sup>A</sup>T<sub>E</sub>X으로 처리하기 위해서는 `xtable` 명령어를 사용하며, 콘솔창에서 `install.packages('xtable')` 명령을 입력하여 설치하고, `library(xtable)` 명령어로 등록을 한 다음, `xtable(모 델 이 름)` 명령어로 사용하면 된다.

SAS에도 Stata와 유사한 기능을 하는 명령어가 있다. 다음 홈페이지를 참고하라(<http://support.sas.com/rnd/base/ods/odsmarkup/latex.html>).

### 10.3 웹의 정보를 활용하여 참고문헌 달기

참고문헌 정보를 일일이 만드는 것도 귀찮은 일 중의 하나이다. 하지만 jstor 등 상당수의 문헌정보 사이트에서 Bib<sub>T</sub>E<sub>X</sub> 출력을 지원한다. 해당 문헌을 소개한 부분을 잘 찾아보면 된다. 경제학의 경우 <http://ideas.repec.org/>에서는 거의 대부분의 문헌에 대한 .bib 파일을 지원하고 있다.

### 10.4 논문용 패키지 활용하기

현재 고려대학교의 경우 공식적인 L<sup>A</sup>T<sub>E</sub>X용 학위논문양식은 없다. KAIST, 한양대, 성균관대 등의 L<sup>A</sup>T<sub>E</sub>X 양식이 있으며 검색을 통해 다운 받을 수 있는 이 양식을 활용하면 고려대학교의 학위논문양식으로 사용할 수도 있다.

논문을 출판하기 위해서는 저널별로 출판 및 투고 양식이 지정되어 있으며, 이에 대해 L<sup>A</sup>T<sub>E</sub>X용 양식 파일을 만들어 두는 경우가 많다. 이러한 패키지 또는 스타일 파일을 저널별로 홈페이지에서 다운 받아 사용하면 된다. 패키지 및 스타일 파일의 설치 및 사용에 대해서는 2장에서 다루었다.