

L^AT_EX 초보의 memoir + memhangul-ucs 연습기록

조인성

2006년 4월 27일

요약

이 문서에서는 memoir와 memhangul-ucs를 연습하는 과정을 담고자 한다. 그렇게 하는 첫째 이유는 memoir를 계속 사용하면서 나중에 필요할 때 개인적으로 참고하기 위해서이고 또 다른 이유는 나의 연습과정 기록이 memoir를 처음 배우고자 하는 사람에게 혹시 도움이 될 수 있을지도 모르겠다는 생각 때문이다. 개인적으로 또는 초보자들이 쉽게 사용하도록 하기 위해, 이 문서에서 표현된 모든 내용의 소스를 소스파일을 보지 않고도 결과 파일에서 직접 볼 수 있도록 했다.

이 문서는 latex 초보자를 위한 것이지만, latex을 처음 사용하는 사람들을 위한 것은 아니다. 이 문서는 latex으로 문서를 만들기 위해서는 `\documentclass`로 시작하고 `\begin{document}`와 `\end{document}` 사이에 본문을 타이핑해야 한다는 것 정도를 책을 보지 않고도 아는 사람을 대상으로 한다.

연습과정의 거의 대부분은 memucs manual(김강수 역)을 따라 한 것에 지나지 않는다. 하지만 몇 군데에서는 manual의 내용과 다르게 표현되기도 했는데, 이는 manual의 내용을 나름대로 이해해서 연습한 것을 적은 것이므로, 잘 못 이해한 채로 적혀 있을 가능성을 배제할 수 없음을 밝혀 둔다. 아울러, 전체 문서의 작성의 내용을 끌어가고 표현하는 데에 있어서도 일관성을 유지하려는 약간의 노력을 기울였으나, 시간제약이 부담이 되거나 게으름이 날 때에는 이 노력을 과감히(?) 포기하였음도 밝혀 둔다. 또한 ‘초보자’의 연습기록이므로, 해당 기능이 memoir 고유의 기능인지 아니면 표준 L^AT_EX클래스에서도 작동하는 것인지를 일일이 구분해낼 수는 없었다는 치명적인 제약도 함께 작용했다는 것도 밝혀 둔다. 가장 부끄러운 것은, 연습하기 위해 인덱스를 몇 개 작성했을 뿐, 진정한 인덱스를 만들지 못했다는 것이다. 이 문서는, 어떤 관점에서 보면, 초보자의 입장을 충실하게 유지하면서 작성된 문서일 수 밖에 없다.

여기는 초록(abstract)이 오는 자리이다. 요즘은 초록이라는 말 대신에 요약이라는 말을 많이 쓴다. 현재 이 문서에도 요약이 디폴트로 쓰였다. 특이한 점은 요약의 첫째 문단에서 indentation이 이루어지고 있다는 점이다.

이 문서는 이번 학기가 시작되기 전까지의 연습기록이고, 학기 시작 후로는 거의 볼 겨를이 없었다. 이 문서를 현재의 상태로 공개하는 것은 내가 원하는 바와는 매우 거리가 먼 일이기 하나, '도은이 아빠'의 지속될 것같은 공개 독촉 압력에 굴복하기로 결정한 것은 내 책임이다.

차례

차례	i
그림 차례	iii
표 차례	iii
제 1 장 단순 챕터 시작하기	3
1.1 절은 어떻게 생겼나? (default: 절까지만 번호 매김)	3
subsection은 어떻게? (default: 소절 이하에는 번호가 붙지 않음)	3
1.2 또 다른 절은 어떻게 생겼나? (장절 번호매기기...)	4
1.2.1 subsection에 번호 붙었나?	4
1.3 Chapter Style 이용하기	4
1.3.1 Chapter 시작하기	4
제 2 장 차례 companion	7
2.1 차례와 면주의 장절표제	7
2.2 chapter style	7
2.3 Some More Variations: Chapter	8
제 3 장 드디어 본문 조판하기	9
3.1 문단모양	9
3.1.1 여러 줄 들여밀기(?) 내어밀기(?)	9
3.1.2 hangfrom: 간단한 내어/들여(?) 밀기 문단	11
3.2 문단폭 바꾸기: adjustwidth 환경	11
3.2.1 adjustwidth	11
3.2.2 adjustwidth*	12
3.2.3 physical paper의 중앙에 위치시키기	12

3.3	나열 문단: itemize, enumerate	13
3.3.1	description: 표준클래스 그대로	13
3.3.2	확장된 itemize	13
3.3.3	확장된 enumerate	14
3.3.4	새로운 list 환경의 정의 (생략가능)	15
제 4 장	frontmatter(ToC etc.) and backmatter(BiB, INDEX)	19
4.1	ToC, LoF, and LoT	19
4.1.1	ToC만들기	19
4.1.2	표제모양 제어하기	20
4.1.3	엔트리 식자	21
4.2	Bibliography	21
4.2.1	thebibliography 환경	21
4.2.2	내게 맞는 bibliography 만들기	22
4.3	Index	24
4.3.1	index 만들기	24
4.3.2	index 만들기: Some tricks	24
제 5 장	캡션과 플롯	27
5.1	캡션	27
5.1.1	캡션 스타일 살짝 바꾸기	28
5.1.2	레전드	30
5.1.3	서브캡션	30
5.2	새로운 플롯 만들기	32
	List of Diagrams	33
제 6 장	행과 열	35
6.1	개요	35
6.1.1	Some Examples	35
6.2	array 환경	38
6.2.1	표(Tables)	39
제 7 장	newenvironment 연습	41
7.1	one example: environment	41
7.2	another example: environment	42

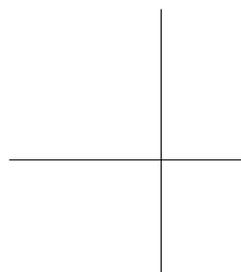
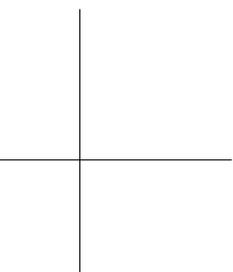
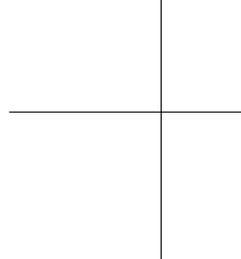
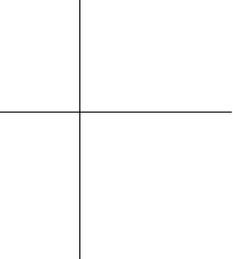
參 考 文 獻	43
찾아보기	45

그림 차례

5.1 밑에 캡션 달기	27
5.2 Prisoners' Dilemma	30
5.3 subcaption test	30
5.4 A figure with some subfigures	31

표 차례

5.1 위에 캡션 달기	27
5.2 Redesigned table caption style	28
5.3	29
6.1 Do not use vertical lines	39



●가정: Windows XP에서 memoir + hlatex-ucs(dhucs) + memhangul-ucs를 설치했다고 하자.

이제, 다음 예제를 foo.tex로 저장하고 latex한다.

```
1 \documentclass{memoir}
2 \usepackage{memhangul-ucs}
3
4 \begin {document}
5 가장 간단한 파일
6 \end{document}
```

```
\marginpar
boxedverbatim
\bvsides
\bvtopandbottom
\nobvbox
\linenumfrequency
\resetbvlینumber
\bvlinenumberinside
\bvlinenumberoutside
\bvbox
```

```
latex foo
```

MiKTeX에 포함된 dvi viewer인 Yap에서 inverse search나 forward search 기능을 이용하려면 --src-specials 옵션을 넣어 latex한다. (어려운 일이 아니므로, 두 번 latex한다.)

```
latex --src-specials foo
```

dvipdfmx도 잘 된다.

역시 간단하지만, 몇 개의 패키지와 옵션을 추가한 파일을 보자.

```
\documentclass[a4paper]{memoir}
%\usepackage{amsmath,amssymb,graphicx}
%%\usepackage[hangul,nonfrench]{dhucs} % memhangul-ucs 안쓰고 한글 쓸 때

%% Using memhangul-ucs
\usepackage[interworddefault,nonfrench,%
             gremph,hangulpagestyle,%
             adjustmath,pdfbookmark]{memhangul-ucs}
\xspaceskip=.7em plus .1em minus .1em %무시해도 됨
\adjustquotespacing %무시해도 됨

\begin {document}
조금 더 복잡하지만 역시 간단한 file.
\end{document}
```

2 표 차례

문서는 `\frontmatter`, `\mainmatter`, `\backmatter`로 구성한다. 위의 예에서 보듯이 꼭 그래야 하는 것은 아니지만 책과 같은 복잡한 문서에서는 이 기능들을 사용하는 것이 편리하다. 이 기능들은 페이지 번호 붙이기, 표제 번호 붙이기 등을 적절하게 조절하는 기능을 포함한다.

boxedverbatim

다음 예는 이러한 구성을 간략히 나타내고 있다.

```
\documentclass{memoir}
\usepackage{memhangul-ucs}

\begin{document}

\title{\bfseries \Huge memoir + memhangul-ucs 연습}
\author{\LaTeX 초보}
\date{\today} %날짜를 생략하려면 \empty

\frontmatter %%%%%%%%%%
\maketitle % 위에서 작성한 타이틀을 출력한다.

\mainmatter %%%%%%%%%%
본문이 오는 곳이다.

\backmatter %%%%%%%%%%
참고 문헌 등이 온다.
\end{document}
```

글을 쓰고 있는 여기는 `mainmatter`에 해당하는 곳이다. `\mainmatter` 선언의 기능은 페이지 번호를 아라비아 숫자로 1부터 시작하도록 하며, 장절명령에 번호를 붙이도록 하는 것 등을 포함한다. 만일 페이지 번호매김과 그 형식에 변화를 주지 않고 지금까지의 형식을 유지하고 싶으면 별표를 붙여 `\mainmatter*`로 선언하면 된다.

`\fancybreak`

`\plainbreak`

아직은 `part`, `chapter`, `section` 등을 시작하지 않은 상태이다. 이제, `chapter`에 관해 살펴보기로 하자.

제 1 장

단순 챗터 시작하기

이 장은 `\chapter{단순 챗터 시작하기}`와 같이 하여 시작했다. 여기는 `chapter`와 `section` 사이이다. 즉 `chapter`는 시작했지만 아직 `section`이 시작되지 않았다. 이제 절을 보기로 하자.

1.1 절은 어떻게 생겼나? (default: 절까지만 번호 매김)

이 절에서는 장절의 번호매김이 디폴트로 되어 있다. 디폴트는 `section`까지만 번호를 매기는 것이다. 즉, 표준클래스에서 다음과 같이 한 것과 같다.

```
\setcounter{secnumdepth}{2} (*)1
```

subsection은 어떻게? (default: 소절 이하에는 번호가 붙지 않음)

이 subsection에는 번호가 붙지 않았다.

subsubsection은 어떻게?

이 subsubsection에서도 번호가 붙지 않았다.

¹(*)표시는 이 표현이 (memoir에서만이 아니라) $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 표준클래스에서도 유효하다는 것을 나타내기 위해 사용되었다. Observe that although we have a footnote before this page, the footnote number starts from one again because this belongs to a new chapter.

1.2 또 다른 절은 어떻게 생겼나? (장절 번호매기기...)

이 절에서는 장절의 번호매김을 subsection까지 붙이기로 한다. 이를 위해서는 다음과 같이 하면 된다.

```
\maxsecnumdepth{subsection}
```

컴파일 중에 \mainmatter 명령이 읽히면, secnumdepth가 \maxsecnumdepth로 설정된다. \maxsecnumdepth 대신 \setsecnumdepth를 써도 된다. 이들 명령은 preamble을 포함하여, 어느 곳이든, 번호를 붙이거나 변경하고자 하는 (sub)section 이전에만 오면 된다.

1.2.1 subsection에 번호 붙었나?

\verbfootnote 이 subsection에는 번호가 붙었다.²

1.2.1.1 subsection에 번호 붙었나?

이 subsection에도 번호가 붙었다. 이 이후로 계속 subsection까지 번호붙이기를 하기로 하자. 맨 마지막에 사용된 \maxsecnumdepth 또는 \setsecnumdepth가 (다시 바뀌기 전까지) 계속 유효하다.

1.3 Chapter Style 이용하기

1.3.1 Chapter 시작하기

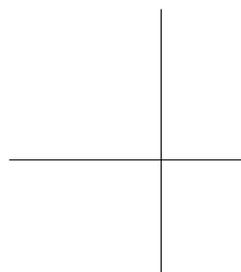
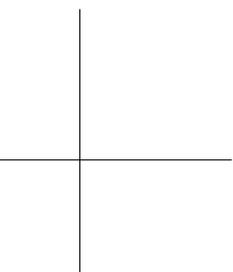
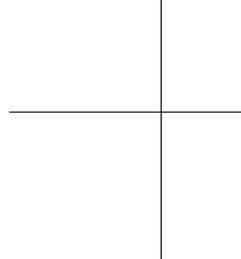
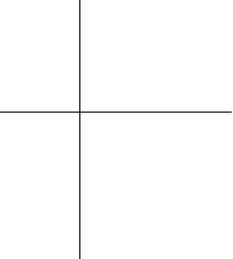
chapter 시작하기는 \openright가 디폴트이며 \opneleft와 \openany도 사용할 수 있다.

이들은 문서클래스의 옵션으로 설정할 수도 있으나, 문서 내 어디에서든 선언하여 사용할 수도 있다. 예를 들면 다음과 같다.

```
\opneany
\chapter{왼쪽 오른쪽 아무 데서나 열리는 챗터}
. . .
```

²특정 장절에 번호를 붙이지 않으려면 \chapter*{...} 또는 \subsection*{...}와 같이 장절명령 뒤에 *를 붙이면 된다. 이 각주 작성은 \verbfootnote{특정 장절에...}와 같이 하여 작성되었는데, 이는 표준적인 \footnote{...}에서와는 달리 각주의 text 안에서 \verb+...+를 쓸 수 있게 한다.

`\cleartorecto`와 `\cleartoverso` 등을 포함한 페이지 이동 명령을 김강수(2006) 『`LATEX`-memoir로 책 만들기』 p.43 에서 보다 상세히 다루고 있다.



제 2 장

companion 챕터 스타일

여기에서는 장절표제형식의 변형에 대해 연습한다. 기본형식(default style)을 이용하면 되므로 시간이 없으면 나중에 연습해도 된다.

* * *

이 장은 다음과 같이 시작하였다.

```
%\chapterstyle{default}          % Why is this used here?
\chapterstyle{companion}
\chapter[차례 companion][면주 companion style]{companion 챕터 스타일}

여기에서는 장절표제형식의 변형에 ...
```

2.1 차례와 면주의 장절표제

`\chapter` 명령은 간단하게는 `\chapter{text}`의 형태로 쓰인다. 이 때 입력된 `text`의 내용이 table of contents와 (짝수쪽) heading에서 사용된다. 만약 `\chapter[option]{text}`와 같이 하면, 차례와 면주에 `option`의 내용이 나타난다.

또한 `\chapter[option 1][option 2]{text}`와 같이 하면, 차례에는 `option 1`, 면주에는 `option 2`의 내용이 나타난다. 실제로, 이 쪽이나 다음의 한두 쪽을 살펴보면 heading에서 ‘면주 COMPANION STYLE’을 볼 수 있을 것이다.

2.2 chapter style

장절표제의 형태를 바꾸려면 default나 companion 등과 같이 미리 정의된 `chapterstyle`을 이용하거나 새로운 `chapterstyle`을 정의하여 사용할 수 있다.

이 전 장은 `\chapterstyle{companion}`를 이용하였고, 이 장에서는 default chapterstyle을 이용하였다. companion chapterstyle은 *The L^AT_EX Companion*의 chapterstyle을 흉내낸 것이라고 한다.

2.3 Some More Variations: Chapter

(to be completed...)

3 드디어 본문 조판하기

이 장은 section 장절 스타일로 조판되었다.

3.1 문단모양

표준 문단의 모양을 제어하는 두 개의 기본적인 매개변수 중 `\parindent`는 문단 첫줄의 들여쓰기 값이고, `\parskip`은 문단과 문단 사이의 수직 간격이다.

이 클래스의 디폴트 `\parindent`는 약 1.52em이며, `\parskip` 값은 일반적으로 0pt이다.

<START—START로 시작하는 이 문단과 END로 끝나는 다음 문단, 두 문단에만 `\parindent` 값이 5em, `\parskip` 값이 10pt가 적용되도록 이 문단 중간에

```
\setlength{\parindent}{5em}와
```

```
\setlength{\parskip}{10pt}를
```

선언하여 해당 값을 조정하였다.

<START에서 END>까지 두 문단을 괄호 {...}로 묶었으므로, 위에서 정한 값은 END 이후 바로 다음으로 이어지는 문단부터는 적용되지 않고, 원래 디폴트로 정해졌던 값이 적용된다. END>

만약 `\parindent`가 음수(-) 값을 갖는다면 문단의 첫번째 줄은 왼쪽 여백을 침범하여 ‘내어밀기’될 것이다.

예로, 이 문단은 `\setlength{\parindent}{-5em}`를 사용하여 여백쪽으로 5em만큼 내어밀기가 되도록 식자되었다. 이 선언은 계속 유효하므로 이 선언에 영향을 받는 범위를 지정해주도록 한다. 간단하게는 해당부분을 괄호 {...}로 묶어주면 된다.

3.1.1 여러 줄 들여밀기(?) 내어밀기(?)

여러 줄 들여/내어밀기는 한 문단의 처음이나 나중의 몇 줄이 문단 내 나머지 행의 길이와 다르게 하는 것이다. 표준 indentation은 처음 한 줄의 길이만 다른 special case로

* * *

이 문단은 `\hangpara{-3em}{-2}`으로 작성되었다. 이 선언은 ‘처음 두행을 왼쪽으로 3em만큼 들여밀라’는 것을 의미한다. 여러 줄 들여밀기 연습문장입니다. 여러 줄 들여밀기 연습문장입니다.

3.1.1.2 hangparas 환경

`hangparas` 환경은 환경내의 모든 문단에 `\hangpara`의 기능이 적용되게 한다. 이를 위해서는 `\begin{hangparas}{indent}{num} ... \end{hangparas}`와 같이 한다.

3.1.2 hangfrom: 간단한 내어/들여(?) 밀기 문단

간단한 내어밀기 문단은 `\hangfromtext` 명령을 사용하면 된다.

* * *

이 문단은 간단한 내어밀기 문단의 예를 보인 것이다. 이 문단과 같은 효과를 내려면 문단을 `\hangfrom{이 문단은 간단}한 내어밀기...` 와 같이 시작하면 된다. 간단하다!

3.2 문단폭 바꾸기: adjustwidth 환경

3.2.1 adjustwidth

문단 폭을 일시적으로 바꿀 때는 `adjustwidth` 환경을 사용하면 편리하다. 이 환경은 다음과 같이 사용한다.

```
\begin{adjustwidth}{<left>}{<right>} ... \end{adjustwidth}
```

`adjustwidth`환경은 환경내 문단에 *left* 길이를 왼쪽 마진에 *right* 길이를 오른쪽 마진에 더해주는 기능을 한다. 이 값이 양(+)*수*이면 문단폭이 줄어들고 음(-)*수*이면 늘어난다. 왼쪽과 오른쪽을 2em만큼 들여쓰게 하면 `quotation`환경과 거의 같은 효과를 낼 수 있다.

3.2.2 adjustwidth*

별표붙은 `adjustwidth*`환경은 *left*를 등(spine)으로, *right*를 배(foledge)로 인식한다. 따라서 홀수쪽에서 두 환경의 효과는 같지만, 짝수쪽에서는 별표(*)가 붙으면 left 길이만큼 등(오른)쪽을 right 길이만큼 배(왼)쪽을 들여쓴다.¹

* * *

예1: 이 문단은 `\begin{adjustwidth}{5em}{2em}`로 시작하여 문단의 왼쪽을 5em, 오른쪽을 2em만큼 들여쓰도록 지정하고, 지금 읽고 있는 문단의 내용을 쓴 후 `\end{adjustwidth}`로 끝나게 식자되었다.

예2: 이 문단은 `\begin{adjustwidth}{2em}{5em}`로 시작하여 문단의 왼쪽을 2em, 오른쪽을 5em만큼 들여쓰도록 지정하고, 지금 읽고 있는 문단의 내용을 쓴 후 `\end{adjustwidth}`로 끝나게 식자되었다.

예3: 이 문단은 `\begin{adjustwidth*}{5em}{2em}`로 시작하여 문단의 등쪽을 5em, 배쪽을 2em만큼 들여쓰도록 지정하고, 지금 읽고 있는 문단의 내용을 쓴 후 `\end{adjustwidth*}`로 끝나게 식자되었다.

* * *

위에 세 문단을 예로 들었다. 별표붙은 환경을 사용한 예3은 어느 페이지에 인쇄되느냐에 따라 그 결과가 다르다. 지금 현재 이 페이지가 홀수이면 예1과 같을 것이고, 짝수이면 예2와 같게 인쇄되었을 것이다. (Am I correct?)

3.2.3 physical paper의 중앙에 위치시키기

문단이나 제목 등을 판면(page)의 중앙이 아니라 인쇄용지의 중앙에 놓이게 하려면 다음과 같이 한다. `\calccentering` 명령과 `adjustwidth*` 환경을 이용하여 간단하게 판 `\calccentering` 면을 용지 중앙에 오게 할 수 있다.²

`\normalfont` 이 문단은 편집영역(판면)이 아니라 용지의 중앙에 오도록 식자되었다. 이를 위해서는 기존에 정해진 임의의 길이—여기서는 `\unitlength`—를 `\calccentering`으로 측정하고 이를 등쪽에 더하고 배쪽에 더해준다.

```
\calccentering{\unitlength}
\begin{adjustwidth*}{\unitlength}{-\unitlength}
```

¹`adjustwidth`환경내에서는 `\noindent` 기능이 꺼지는 것으로 보인다.

²This technique does not seem to be robust.

```
...
\end{adjustwidth*}
```

3.3 나열 문단: itemize, enumerate

3.3.1 description: 표준클래스 그대로

```
\begin{description}
  \item[description] 표준 \LaTeX에서와 같이 memoir에서도 ...
  \item[itemize] ...
  \item[enumerate] ...
\end{description}
```

description 표준 \LaTeX 에서와 같이 memoir에서도 description환경을 제공한다. 표준 \LaTeX 에서와 같이 memoir에서도 description환경을 제공한다.

itemize 환경 itemize에서는 확장된 기능을 제공한다. 환경 itemize에서는 확장된 기능을 제공한다. 환경 itemize에서는 확장된 기능을 제공한다.

enumerate 환경 enumerate에서도 확장된 기능을 제공한다. 환경 enumerate에서도 확장된 기능을 제공한다. 환경 enumerate에서도 확장된 기능을 제공한다.

3.3.2 확장된 itemize

```
\begin{itemize}[<mark>] \item ... \end{itemize}
```

*mark*의 디폴트값은 \bullet `\textbullet`이고 그 밖에 어떤 것도 올 수 있다. 예로,

`\tightlist`

- `– \textendash`,
- `* \textasteriskcentered`,
- `→ \rightarrow`,
- 또는 `¶ \P(pilcrow)` 등을 들 수 있다.

위의 예는 다음과 같이 하여 작성되었다.

```
\begin{itemize}[\bfseries\textperiodcentered]\tightlist
  \item \textendash\, \verb+\textendash+,
  \item \textreferencemark\, \verb+\textreferencemark+
  ...
\end{itemize}
```

`\tightlist``\firmlist``\tightlists``\defaultlists`

위의 예에서, 환경시작 직후에 사용된 `\tightlist`는 환경내의 나열항목 사이의 수직간격을 없애주는 역할을 한다. `\firmlist`는 조금 줄어든 수직간격을 얻게 한다. 어떤 때는 chapter 전체에서 매환경마다 `\tightlist`를 넣어야 할 필요가 있을 수도 있다. 이 때에는 chapter 초기에 (또는 환경시작 전에) `\tightlists`를 선언하면, 이 선언 이후의 모든 환경내에 `\tightlist`를 넣은 것과 같은 결과를 얻을 수 있다. 항목 사이의 간격을 표준대로 넓게 하고 싶으면 `\defaultlists`를 선언하면 된다.

3.3.3 확장된 enumerate

```
\begin{enumerate}[<style>] \item ... \end{enumerate}
```

`style`의 디폴트값은 ‘1.’(즉, 아라비아숫자 + period)이다. 아라비아 숫자 1은 첫째 항목의 번호이며 그 후의 항목번호는 같은 스타일을 유지하면서 자연스럽게 증가된다. `style`은 1), [A], a., I. (i)등 여러가지 형태로 나타낼 수 있다.

`style` 옵션의 값에 숫자와 알파벳이 아닌 문자(예로, `\textbullet`)가 오면 `itemize` 환경의 글머리부호 `mark`를 `\bullet`으로 지정한 것과 같은 효과를 가져온다.

`style` 옵션에 한글 ‘가.’를 넣어도 문자로 간주되어 모든 항목의 번호가 ‘가. 가. ...’ 처럼 표시된다. 항목 번호를 ‘가. 나. 다. ...’와 같이 표시하려면 다음과 같이 한다.³

- 가. 이 예에서는 `[style]` 옵션 대신에 `\renewcommand{\labelenumi}{\gana{enumi}.}`를 사용하였다.
- 나. `\gana` 대신에 `\ogana`를 사용하면 ㉠ ㉡ ㉢...와 같이 된다.
- 다. `\pjaso`를 사용하면 ㉠ ㉡ ㉢...과 같이 되고
- 라. `\onum`를 사용하면 ① ② ③...과 같이 된다.

³이 방법은 L^AT_EX의 표준클래스에서 `enumerate` 환경의 numbering style을 바꿀 때 쓰는 방법이다. `\labelenumi` 대신에 `\labelitemi`를 쓰면 `itemize` 환경의 글머리부호도 바꿀 수 있으나, memoir클래스에서 글머리부호 옵션인 `mark` 옵션을 사용하는 것이 보다 편리하다 하겠다.

3.3.4 새로운 list 환경의 정의 (생략가능)

바쁜 사람들은 이 subsection을 생략하고 다음으로 넘어가도 된다.

3.3.4.1 새로운 환경

```
\begin{list}{<default-label>}{<code>} \item ... \end{list}
```

예를 통해 보기로 하자.

```
%%% 항제목을 이탤릭체로 하고 항목간 수직간격을 줄인 description 환경
\newcommand{\itlabel}[1]{\hspace\labelsep\normalfont\itshape #1}
\newenvironment{itdesc}%
  {\list{}{%
    \setlength{\labelsep}{0.5em}
    \setlength{\itemindent}{0pt}
    \setlength{\leftmargin}{\parindent}
    \setlength{\labelwidth}{\leftmargin}
    \addtolength{\labelwidth}{-\labelsep}
    \setlength{\listparindent}{\parindent}
    \setlength{\parsep}{\parskip}
    \setlength{\itemsep}{0.3\onelineskip} %항목간 수직간격 조절
    \let\makelabel\itlabel}}%
  {\endlist}
%%% 사용하는 것은 보통의 description환경과 같다.
\begin{itdesc}
  \item[italic]
  ...
\end{itdesc}
```

italic This item *label* is printed in italic shape. 이탤릭체로 하는 것이 표준적인 description환경의 표제보다 보기 좋은가?

description 전 항과 이 항의 항목간 간격은 tighter해졌나?

```
%% glossary list environment 정의하기
%% 다음과 같이 본문 또는 preamble에 정의한다.
\newenvironment{aglossary}%
```

```

{\begin{list}}{% empty label
  {\setlength{\topsep}{\baselineskip}
   \setlength{\partopsep}{0pt}
   \setlength{\itemsep}{0.5\baselineskip}
   \setlength{\parsep}{0pt}
   \setlength{\leftmargin}{2em}
   \setlength{\rightmargin}{0em}
   \setlength{\listparindent}{1em}
   \setlength{\itemindent}{0em}
   \setlength{\labelwidth}{0em}
   \setlength{\labelsep}{2em}}}%
{\end{list}}
\newcommand{\gloss}[1]{\item[\bfseries #1]\mbox{}\nopagebreak}

%% aglossary 환경 사용하기
\begin{aglossary}
  \gloss[function] A function from ...
  \gloss[한계대체율] ...
\end{aglossary}

```

위의 coding의 예를 따라 다음 나열문단을 작성하였다.

function A function from X to Y is a rule that assigns to each element in X one and only one element in Y .

한계대체율 한계대체율이란 동일한 만족수준을 유지하면서 X 재 한 단위를 더 소비하기 위해 포기해야 하는 Y 재의 양을 말한다.

3.3.4.2 환경 전후의 수직간격 조정

`center` 등의 `trivlist` 환경들을 사용하면 전후에 약간의 수직간격이 더해진다. 이 간격을 없애려면 환경시작 (바로) 전에 `\zerotrivseps`를 선언하라.

이 문단은 `center`환경을 이용하여 작성였다. 이 문단 전후에 추가되는 수직간격을 없애기 위해 환경시작 전에 `\zerotrivseps`를 선언했는데 간격이 없어졌나?

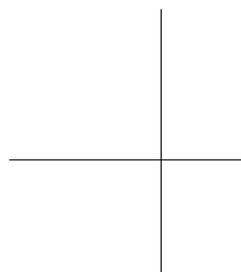
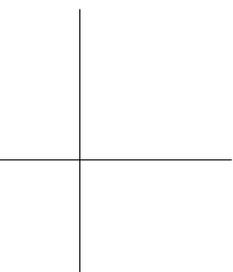
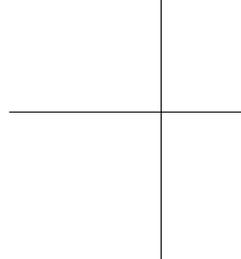
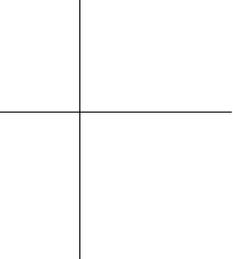
환경내에서 선언하면 원하는 결과를 얻을 수 없다. 환경 전후의 추가되는 수직간격을 원래대로 되돌리려면 `\restoretrivseps`를 선언하면 된다.

이 문단은 *center* 환경을 이용하여 작성하였다. 이 문단 전후에 추가되는 수직간격을 회복하려고 환경시작 전에 `\restoretrivseps`를 선언했는데 간격이 돌아왔나?

간격이 (원래대로) 회복되었나? `\zerotrivseps`를 선언하는 것은 다음에서 의미하는 것과 같은 효과를 얻는다.

```
\renewcommand{\zerotrivseps}{%  
\setlength{\topsep}{0pt}%  
\setlength{\partopsep}{0pt}}
```

필요하다면 0pt 부분을 조정하여 fine tuning을 할 수 있을 것이다.



4 frontmatter(ToC etc.) and backmatter(BiB, INDEX)

이 장은 hangnum 챗터스타일로 조판되었다. `\hangsecnum` 을 선언하면 그 선언 이후의 절번호는 여백으로 빠져나간다.

4.1 ToC, LoF, and LoT

4.1.1 ToC만들기

memoir 클래스는 `\tableofcontents` 나 `\tableofcontents*`가 선언된 위치에서 ToC를 만든다. 일반적으로 `\tableofcontents`를 선언하는 곳은 `\mainmatter`를 선언하기 직전이면 된다. 하지만 바로 이곳에 `\tableofcontents`를 선언하면 바로 이곳에 ‘차례’가 만들어진다. (Try it yourself.)

YOU WANT ToC RIGHT HERE?

마찬가지로 `\listoffigures` 나 `\listoffigures*`가 선언된 위치에서 LoF가 만들어지고, `\listoftables` 나 `\listoftables*`가 선언된 위치에서 LoT가 만들어진다. 별표(*) 붙은 선언은 자신의 heading을 ToC에 추가하지 않는다.

표준 L^AT_EX 클래스에서와는 달리 memoir 클래스에서는 ToC, LoF, LoT를 새로운 페이지에서 시작하지 않는다. 새로운 페이지에서 시작하고 싶으면 `\cleardoublepage`나 `\clearpage`를 이용하면 된다. `\tableofcontents`를 선언하기 전에 `\chapterstyle{...}` 명령을 선언하여 챗터스타일을 조절할 수도 있다.

어떤 장절까지 ToC 엔트리에 넣을까 하는 것은 `\maxtocdepth{...}`를 사용하거나 `\settocdepth{...}`를 사용하여 조절한다. 기본값은 `\maxtocdepth{section}`이다. 이 명령은 preamble을 포함하여 문서 내 어느 곳에 와도 된다.

```

...
\maketitle
...
\cleardoublepage
\chapterstyle{demo}

\maxtocdepth{subsection} %생략하면 section까지 표시
\tableofcontents*
\clearpage
\listoffigures
...
\mainmatter %%%%%%%%%%

```

4.1.2 표제모양 제어하기

초보자가 표제모양을 제어할 필요가 있을까? 표제모양 제어의 필요성은 초보자인지의 여부와는 관계없는 문제이다. 문제는 초보자가 표제모양 제어 방법을 배울 필요가 있는 것인가이다. 지금 당장 필요하지 않으면 다음으로 미루는 것도 좋을 것이다.

본질적으로 ToC, LoF, LoT 표제는 chapter표제와 동일한 형식을 가지므로 현재의 chapterstyle에 의해 모양이 결정된다. 이 모양을 바꾸는 예를 몇 개 들기로 하자. toc를 lof나 lot로 바꾸어도 같은 논리가 성립한다.

* 표제가 오른쪽 정렬되어 Large italic 폰트로 식자되게 하려면:

```
\renewcommand{\printtoctitle}[1]{\hfill\Large\itshape #1}
```

* 표제가 Large bold로 중앙정렬되기를 원하면:

```
\renewcommand{\printtoctitle}[1]{\centering\Huge\bfseries #1}
```

* ToC의 첫 페이지는 empty 페이지 스타일이 되게 하려면:

```
\renewcommand{\aftertoctitle}{\thispagestyle{empty}\afterchaptertitle}
```

* ‘Page’라는 단어를 표제가 앉혀진 줄에 flushright되게 하려면:

```
\renewcommand{\aftertoctitle}{%
\par\nobreak \mbox{}\hfill{\normalfont Page}\par \nobreak}
```

\printtoctitle은 실제로 표제를 식자한다. (\printchaptertitle이 기본으로 설정되어 있다.) \aftertoctitle은 표제가 식자된 후에 불린다. (\afterchaptertitle이 기본으로 설정되어 있다.)

4.1.3 엔트리 식자

ToC 엔트리와 페이지를 연결하는 줄은 기본적으로 `\renewcommand{\cftdot}{.}`로 설정되어 있다. `\renewcommand{\cftdot}{\ensuremath{\ast}}`로 하면 점선이 아니라 *로 줄을 긋는다.

이 이외에 많은 조절기능이 있으나, 나중에 살펴보기로 하자.

4.2 Bibliography

4.2.1 thebibliography 환경

참고문헌 목록 또는 서지(bibliography)를 만드는 명령은 표준 L^AT_EX클래스의 것과 동일하지만 memoir 클래스에서 그 기능이 개선되었다. 색인(index)의 경우도 L^AT_EX의 명령과 동일하지만 그 기능이 현저히 개선되었다. bibliography와 index는 `\backmatter` 후에 온다.

```
\begin{thebibliography}{<exlabel>}
  \bibitem [<x>]{StewartCal} Stewart, James, \textit{Calculus}, ...
  \bibitem ...
\end{thebibliography}
\bibname
```

bibliography는 thebibliography환경으로 만든다. 이 환경이 취하는 인자 *exlabel*은 bibliography에서 가장 넓은 label의 폭을 나타내기 위해서 주는 문자이다.¹ `\bibname`의 값이 표제로 사용되는데, 기본값은 ‘Bibliography’이다. `memhangul-ucs`에서 `\bibname`의 기본값은 ‘참고 문헌’이다. `\renewcommand{\bibname}{參考文獻}`을 bibliography 환경 전에 식자하면 표제를 ‘참고문헌’으로 바꿀 수 있다.

```
\renewcommand\prebibhook{...}
\renewcommand\postbibhook{...}

\setbiblabel{\textperiodcentered}
```

¹통상적으로 참고문헌이 9개 이하이면 *exlabel*을 문자 ‘9’로 주고, 99개 이하이면 문자 ‘99’로 준다. 이 경우 문자 ‘99’가 인자되는 폭이 label의 최대폭이 되는 것이다. 문자 ‘ii’의 폭은 더 좁을 것이고 문자 ‘MM’의 폭은 넓을 것이다. `\bibitem`의 두 인자 [...]와 {...}는 L^AT_EX클래스에서와 같은 의미를 가지므로 무슨 뜻인지를 적절한 문헌을 통해 알아볼 것을 권장한다.

```

\renewcommand{\bibname}{参考文献}
\begin{thebibliography}{99}
  \bibitem{HPW} Haeussler..., \textit{Introductory...}, ...
  \bibitem{Sydsaeter} Sydsaeter..., \textit{Essential...}, ...
  \bibitem ...
\end{thebibliography}

```

`\prebibhook{...}`과 `\postbibhook{...}`을 `bibliography` 환경 전에 두면 `{...}`의 내용을 각각 문헌목록의 첫 항목 전과 마지막 항목 후에 식자한다. `\nobibintoc` 선언은 ToC에 표제 ‘Bibliography’가 나타나지 않게 한다. 기본값은 표제가 나타나도록 하는 `\bibintoc`로 설정되어 있다.

위와 같이 식자하면 참고문헌은 다음과 같이 (유사하게) 나타난다.

- [1] Haeussler, Ernest F., Richard S. Paul, and Richard Wood, *Introductory Mathematical Analysis*, 11th ed., 2003.
- [2] Hughes-Hallett, Deborah, Andrew M. Gleason, and William G. McCallum, *Calculus: Single and Multivariable*, 4th ed., 2005.
- [3]

참고문헌 항목간의 수직 간격은 `\bibitemsep`으로 조절할 수 있다. `\bibitemsep`의 기본값은 `\itemsep`의 값으로 지정되어 있다. 참고문헌은 `list`로 식자되므로, 항목간의 간격은 `\bibitemsep + \parsep`이다. `\setlength{\bibitemsep}{-\parsep}`을 하면 항목간 추가 간격을 제거할 수 있다. 그 결과는 다음과 같다.

- [1] Haeussler, Ernest F., Richard S. Paul, and Richard Wood, *Introductory Mathematical Analysis*, 11th ed., 2003.
- [2] Hughes-Hallett, Deborah, Andrew M. Gleason, and William G. McCallum, *Calculus: Single and Multivariable*, 4th ed., 2005.
- [3]

4.2.2 내게 맞는 `bibliography` 만들기

다음과 같이, 번호 없이 들여쓰기만 되는 형태를 만들려면 어떻게 해야 하나?

Haeussler, Ernest F., Richard S. Paul, and Richard Wood, *Introductory Mathematical Analysis*, 11th ed., 2003.

Hughes-Hallett, Deborah, Andrew M. Gleason, and William G. McCallum, *Calculus: Single and Multivariable*, 4th ed., 2005.

.....

한 가지 방법으로 다음과 같이 fine-tuning하는 방법을 생각해 볼 수는 있겠다. 먼저 새로운 길이변수 `\mybibindent`를 정의하여 들여쓰기의 길이를 정의하는 것이 편리하다. 다음으로 번호를 없애는 것은 `\setbiblabel{}` 또는 `\setbiblabel{\empty}`을 명령하면 된다. 다음으로 `\biblistextra`를 재정의하여 원하는 문헌목록 형태를 위와 유사하게 만들 수 있다. `\biblistextra`의 기본값은 아무 것도 하지 않는 것이지만, 재정의를 통해 list 파라미터를 수정하는 데 사용된다.

단, `\biblistextra`를 이용하면 `\setlength{\bibitemsep}{-\parsep}`이 작동하지 않는다(I don't know why). 따라서 이를 위한 추가적인 조정이 필요하다. 여기서는 인위적으로 `\vspace{-\parsep}`를 넣어 조정했다. 이 두가지 일을 한꺼번에 하기 위해 임시로 새로운 명령 `\mybibstyle`을 정의하여 썼다.²

```
\newlength\mybibindent
\setlength\mybibindent{1.5em}

\renewcommand{\biblistextra}{%
  \setlength{\leftmargin}{\mybibindent}%
  \setlength{\itemindent}{Opt}%
}

\newcommand\mybibstyle{\vspace{-\parsep}\hspace{-\mybibindent}} %%%
\setlength{\bibitemsep}{-\parsep} %does not work with "\biblistextra"
\setbiblabel{}
\begin{thebibliography}{}
  \bibitem{HPW}\hspace{-\mybibindent}Haeussler, .....
  \bibitem .....
\end{thebibliography}
```

²`\mybibindent`와 `\mybibstyle` 명령은 사용자가 임시 방편 목적으로 정의하여 쓴 것이다.

정리 4.3 Index

makeindex.exe

정리! 정점정리
정리! 재고 정리

4.3.1 index 만들기

이 클래스는 `makeidx`, `showindex`, `index` 패키지에서 제공하는 기능을 구현하므로, 이 패키지들과 함께 쓰면 안된다.

단어 ‘정리’를 인덱스에 포함시키려고 한다면 어느 곳이나 `\index{정리}`와 같이 명령하면 이 단어가 본문에는 나타나지 않고 인덱스 엔트리로 나타난다.

`\makeindex`를 preamble에 선언하면 `index`의 input파일인 `foo.idx`가 만들어진다. 이 `foo.idx`를 `index`의 output 파일인 `foo.ind`로 바꾸어주는 것은 `makeindex.exe`이다. (보통은 bibliography 이후에) `\printindex`를 선언하여 output 파일 `foo.ind`를 읽어 들임으로써 `index`를 만든다.

따라서, `index`를 포함한 결과를 얻으려면 다음과 같은 순서로 컴파일해야 한다.³

```
latex foo.tex    %\makeindex 선언으로 idx파일 만들
makeindex foo.idx %idx파일을 ind파일로 바꿈
latex --src-specials foo.tex %\printindex 선언으로 ind파일을 읽음
```

인덱스 표제는 `\indexname`의 값이다. memoir에서 초기값은 ‘Index’이며 `memhangu-ucs`에서의 초기값은 ‘찾아보기’이다. 만약 어떤 이유에서든 인덱스의 표제를 ‘찾아서 보기’로 바꾸고 싶다면 `\renewcommand\indexname{찾아서 보기}`와 같이 하면된다.

4.3.2 index 만들기: Some tricks

‘정리’라는 큰 엔트리에 속하는 작은 엔트리로 정점정리와 재고정리를 넣고 싶으면 다음과 같이 한다.

```
‘정리’라는 큰 엔트리에 속하는 작은 엔트리로
\index{정리!정점정리}정점정리와
\index{정리!재고 정리}재고정리를 넣고 싶으면...
```

어떤 단어를 `index` 엔트리로 지정하면 본문에는 나타나지 않으므로 그 단어를 중복해서 `type`해야 하는 것이 비효율적이라고 생각한다면 다음 예를 보라!

³이 내용을 `golatex.bat`으로 만들어 사용하면 편리하다.

```

\newcommand\windex[2][\empty]{%
  \ifx#1\empty
    \index{#2}#2%
    \PrerenderUnicode{#2}%
  \else
    \index{#1!#2}#2%
    \PrerenderUnicode{#1#2}%
  \fi
}

```

\windex[정리]{점포정리}도 간단하게 인덱스의 작은 엔트리로 ...

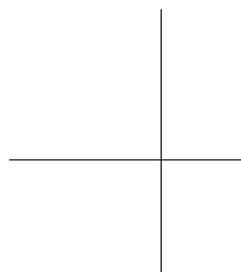
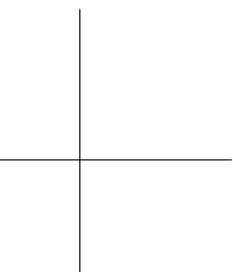
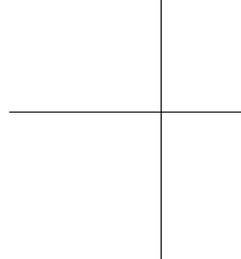
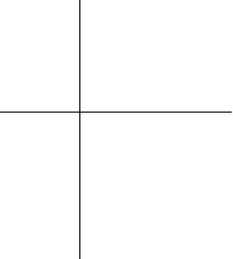
정리!점포정리

위와 같이 하면 ‘점포정리’를 한 번만 type하여 인덱스의 작은 엔트리로 포함시킬 수 있다.⁴

인덱스에 포함된 단어를 해당페이지의 여백에 표시하면 인덱스 관리에 편리할 것이다. 이를 위해서는 (preamble에) \showindexmarktrue를 선언하면 된다. 최종 인쇄물에서 여백의 인덱스를 없애려면, 이를 comment 처리하면 된다. 간단하다.

\showindexmarktrue

⁴새로운 명령어 \windex를 정의하는 이 코드는 김강수(2004)에서 가져 온 것이다.



5 캡션과 플롯

이 장은 (default 캡터스타일을 먼저 부른 후) hangnum 캡터스타일로 조판되었다. 여기에서는 `\hangsecnum` 을 선언하지 않았지만 이미 4.1.1에서 선언되었기 때문에 계속 장절번호가 왼쪽 마진까지 나온다.¹

5.1 캡션

memoir에서는 캡션의 기능이 대폭 확장되었다. 먼저, 이 클래스에서의 기본값이 적용된 캡션의 모습을 보자. 그림이나 표의 내용 앞에 `\caption{...}`을 달면 그림이나 표 위에 캡션이 나오고, 아래에 달면 밑에 나온다.

figure here

그림 5.1: 밑에 캡션 달기

표 5.1: 위에 캡션 달기

table here

```
\begin{figure}[h]
  \centering \fbox{figure here} %내용
  \caption[밑의 캡션]{밑에 캡션 달기}
\end{figure}

\begin{table}[h]
\caption{위에 캡션 달기}
  \centering \fbox{table here} %내용
\end{table}
```

¹어떻게 해야 `\hangsecnum`의 효과를 없앨 수 있을까?

5.1.1 캡션 스타일 살짝 바꾸기

캡션의 기본적인 형태는 [캡션이름] + [분리자] + [캡션 제목]으로 이루어져 있다. 이들의 모양을 원하는대로 바꿀 수 있다. 예를 들어, 다음과 같이 바꿀 수 있다. 이들을 플롯 환경 안에 코딩하면 그 플롯 환경에만 유효하고 밖에 하면 그 이후의 모든 플롯 환경에 영향을 미친다.

```
\captionnamefont{\Large\ssfamly} %default: {}
\captiondelim{-- } %default: {: }
\captiontitlefont{\itshape} %default: {} for normal
```

`\precaption{...}`과 `\postcaption{...}` 명령은 각각 캡션 전후에 처리된다. 이를 이용하여, 캡션 위에 선을 그으려면 `\precaption{\rule{\linewidth}{0.8pt}\par}`로 코딩하고 `\postcaption{\rule{\linewidth}{0.4pt}}`로 하면 캡션 아래에 선이 그려진다. 이 두 명령은 플롯 내에서 쓰이면 이 명령 다음에 오는 모든 캡션에 영향을 미친다. 그러나, 프리앰블이나 본문에 오면 그 효과는 파라미터 값을 바꾸기 전까지 계속된다.

결국, 캡션은 다음과 같은 형태로 식자된다.

```
\precaption
{\captionnamefont NAME NUMBER \captiondelim}
{\captionstyle\captiontitlefont THE TITLE}
\postcaption
```

`\captionstyle[...]{...}` 선언을 이용하여 캡션의 좌우정렬, 가운데 정렬 등을 조절할 수 있다. 아래의 예에서는 `\captionstyle{\}`이 이용되었다. 이는 캡션이름과 캡션타이틀의 줄을 나누는 스타일이다.

표 5.2
REDESIGNED TABLE CAPTION STYLE

Some Table Here!

표 5.2의 코딩은 다음과 같다.

```
\begin{table}[htb]
\centering
```

```

\captionnamefont{\sffamily}
\captiondelim{}
\captionstyle{\}
\captiontitlefont{\scshape}
\setlength{\belowcaptionskip}{10pt}
\caption{Redesigned table caption style} \label{tab:style}
\fbbox{Some Table Here!}
\end{table}

```

캡션 스타일을 preamble이나 .sty 패키지 파일에 적어 놓고 본문에는 간략하게 코딩하려면 다음과 같이 할 수 있다.

```

\makeatletter
\newcommand{\mycaption}[2][\@empty]{
\captionnamefont{\sffamily\hfill}
\captiondelim{\hfill}
\captionstyle{\centerlastline\}
\captiontitlefont{\scshape}
\setlength{\belowcaptionskip}{10pt}
\ifx #1 \@empty \caption{#2}
\else \caption[#1]{#2}
\makeatother

```

먼저 위의 코딩을 preamble에 둔다 (\centerlastline의 효과에 대해서는 매뉴얼을 참고하라). 그리고, 다음을 코딩하면 표 5.3을 얻는다.

```

\begin{table}[htb]
\centering
\mycaption{Redesigned table caption style: again} \label{tab:style1}
\fbbox{Some Table Here!: Again}
\end{table}

```

표 5.3

REDESIGNED TABLE CAPTION STYLE: AGAIN

Some Table Here!: Again

5.1.2 레전드

`\legend{...}` 명령은 플롯 내에서 `\caption` 명령과 무관하게 사용할 수 있다.

	<i>C</i>	<i>D</i>
<i>C</i>	2,2	0,3
<i>D</i>	3,0	1,1

그림 5.2- Prisoners' Dilemma

```

\begin{table}[htbp]
\captiondelim{-- } % 캡션 제목 분리자 바꾸기
\legend{simple version}
\begin{center}
\begin{tabular}{|c|c|c|}\hline
&  $C$  &  $D$  \\ \hline
 $C$  & 2,2 & 0,3 \\ \hline
 $D$  & 3,0 & 1,1 \\ \hline
\end{tabular}
\end{center}
\legend{below}
\end{table}
\caption{Prisoners' Dilemma}

```

`\caption`과 마찬가지로 레전드 전후의 공백은 `\abovecaptionskip`과 `\belowcaptionskip`의 길이로 조절한다.

그 밖에 `legend`에 관한 기능이 많이 있다. (to be completed...)

5.1.3 서브캡션

`memoir` 클래스에서 서브캡션을 구사하려면 `subfigure` 패키지를 불러서 함께 쓴다. 그러나 이 클래스에서는 기능을 확장하였다.

subfigure
패키지

subfigure ONE

(a) subcaption 1

subfigure TWO

(b) subcaption 2

그림 5.3: subcaption test

그림 5.3에서는 `\subcaption{... \label{...}}`과 같이 함으로써 그림 5.3(a)와 (b) `\subcaptionref`를 얻었다.

```

\begin{figure}[htb]
\centering
\begin{minipage}{0.3\textwidth}
\fbbox{subfigure ONE}
\subcaption{subcaption 1\label{sf:sc1}}
\end{minipage}
\hspace{3em}
\begin{minipage}{0.3\textwidth}
\fbbox{subfigure TWO}
\subcaption{subcaption 2\label{sf:sc2}}
\end{minipage}
\caption{subcaption test}\label{fig:subcaption}
\end{figure}

```

그림 `\ref{fig:subcaption}` 에서는 `\verb+\subcaption{...\label{...}}+` 와 같이 함으로써 그림 `\ref{sf:sc1}` 과 `\subcaptionref{sf:sc2}` 를 얻었다.

`\subcaption`은 고정폭 환경에서 서브캡션을 넣는다. 한편, `\subtop`과 `\subbottom`은 내용의 폭을 이용하여 서브캡션을 식자한다. 두 그림의 서브캡션의 위치를 비교해보라! 그림 5.4는 세 개의 subfigure를 가지고 있는데, 바로 5.4(a)와 (b)와 (c)다.

`\dotemph`
`\circemph`
`\subtop`
`\subbottom`

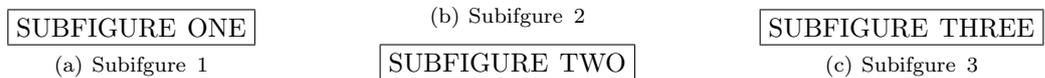


그림 5.4: A figure with some subfigures

그림 `\ref{subfig:sf}`는 세 개의 subfigure를 가지고 있는데, 바로 `\ref{sf:1}`과 `\subcaptionref{sf:2}`과 `\subcaptionref{sf:3}`이다.

```

\begin{figure}[htb]
\centering
\subbottom[Subifigure 1]{\fbbox{SUBFIGURE ONE}\label{sf:1}}
\hfill
\subtop[Subifigure 2]{\fbbox{SUBFIGURE TWO}\label{sf:2}}
\hfill
\subbottom[Subifigure 3]{\fbbox{SUBFIGURE THREE}\label{sf:3}}
\caption{Figure with some subfigures} \label{subfig:sf}
\end{figure}

```

5.2 새로운 플롯 만들기

다음 예를 보자.

먼저 프리앰블에 다음과 같이 넣는다.

```
\newcommand{\diagramname}{Diagram}
\newcommand{\listdiagramname}{List of Diagrams}
\newlistof{listofdiagrams}{dgm}{\listdiagramname}
\newfloat{diagram}{dgm}{\diagramname}
\newfixedcaption{\fdiagcaption}{diagram}
\newlistentry{diagram}{dgm}{0}
```

그리고 본문에 다음의 내용을 식자한다.

Diagram 1: A diagram

A diagram

You can see diagram 1 and diagram 2, which are just to show examples of using new float environment.

Diagram 2: Another diagram

Another diagram

diagram 1과 diagram 2는 다음과 같이 코딩하였다.

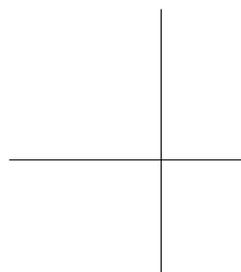
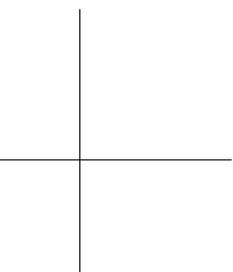
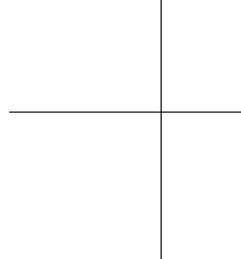
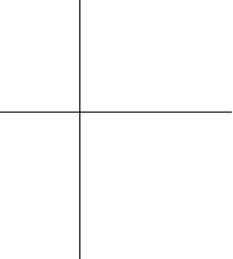
```
\begin{diagram}[htb]
\caption{A diagram} \label{diag1}
\centering \fbox{Diagram 1}
\end{diagram}
You can see diagram~\ref{diag1} and diagram~\ref{diag2}, which are
just to show examples of using new float environment.

\begin{minipage}{.9\textwidth}
\fdiagcaption{Another diagram} \label{diag2} %%
\centering \fbox{Diagram 2}
\end{minipage}
```

원하는 곳에 `\listofdiagrams`를 두면, 그 곳에 ‘List of Diagrams’가 식자된다.

List of Diagrams

1	A diagram	32
2	Another diagram	32



제 6 장

행과 열

이 장은 section 챗터스타일로 조판되었다. 여기에서도 `\hangsecnum` 을 선언하지 않았지만 이미 앞에서 선언되었기 때문에 장절번호가 여전히 왼쪽 마진까지 나온다.¹ 이 장에서는 `array`와 `tabular`환경 기능의 확장을 공부한다.

`\verbfootnote`

`\footref`

6.1 개요

```
\[ begin{array}[<pos>]{<preamble>} ... \end{array} \  
\begin{tabular}[<pos>]{<preamble>} ... \end{tabular}  
\begin{tabular*}[<width>][<pos>]{<preamble>} ... \end{tabular*}  
\begin{tabularx}[<width>][<pos>]{<preamble>} ... \end{tabularx}
```

`<pos>` 옵션인자는 `t(op)`, `c(enter)`, `b(ottom)` 중의 하나이며, 기본 값은 `c(enter)`이다. 열의 개수와 형식은 `<preamble>`에 의하여 지정된다. `tabular*`와 `\tabularx`는 `<width>` 인자로 폭의 길이를 제어할 수 있다.

6.1.1 Some Examples

성명	번호	비고
Abraham	305	월드컵 관전
Bob	200512	2번 마을 버스

¹어떻게 해야 `\hangsecnum`의 효과를 없앨 수 있을지는 여전히 의문이다. 각주²에서도 말했지만, 이 각주와 같이 memoir의 `\verbfootnote{...}` 명령을 사용하여 식자하면, 각주 내에서 `\verb+\hangsecnum+`를 바로 쓸 수 있으므로 편리하다! 여기서 각주번호 상호참조는 `\footref{<label>}`으로 식자했다.

```
\begin{center}
\begin{tabular}{>{\large\bfseries}r|c >{\small\itshape}l} \toprule
성명 & 번호 & 비고 \\ \midrule[0.2pt]
Abraham & 305 & 월드컵 관전 \\
Bob & 200512 & 2번 마을 버스 \\ \bottomrule
\end{tabular}
\end{center}
```

b는 bottom 에 정렬하도 록 한다.	m은 baseline 을 기준으로 middle에 ...	p는 (t가 아님에 주목) <code>\parbox[t]{<width>}</code> 와 동일하다.
------------------------------	--------------------------------------	---

```
\begin{center}
\begin{tabular}{b{2cm}|m{2cm}|p{4cm}}\toprule
b는 bottom에 정렬하도록 한다. & & \\
m은 baseline을 기준으로 middle에 ... & & \\
p는 (t가 아님에 주목) & & \\
\verb+\parbox[t]{<width>}+와 동일하다. \\ \bottomrule
\end{tabular}
\end{center}
```

`\tabularnewline`

0,1	-1,3	1,3
1,1	3,1	1,2
0,1	1,3	1,0

여기에서 `\\`를 사용하지 않고(why? 작동하지 않아서) `\tabularnewline`을 사용하여 행을 바꾸었다. `\\`가 작동하지 않으면 `\tabularnewline`를 사용하자.

```
\begin{center}\marginpar{\tabularnewline}
\begin{tabular}{|>{\centering$}m{3em}<{\$}%
|>{\centering$}m{3em}<{\$}%
|>{\centering$}m{3em}<{\$}|} \hline
0,1 & -1,3 & 1,3 \tabularnewline \hline
1,1 & 3,1 & 1,2 \tabularnewline \hline
\end{tabular}
```

```
0,1 & 1,3 & 1,0 \tabularnewline \hline
\end{tabular}
\end{center}
```

새로운 컬럼 지시자를 정의해서 사용하면 편리하다.

`\newcolumntype`

```
\newcolumntype{M}{>\centering$m{3em}<{}}
```

컬럼 지시자를 `\newcolumntype{M}{>{<}>\centering$m{3em}<{}}`와 같이 정의해도 마찬가지로 결과를 얻을 수 있다.

0,1	1,-3	1,3
1,1	3,1	-1,2
0,1	1,3	1,0

```
\begin{center}
\begin{tabular}{|M|M|M|} \hline
0,1 & 1,-3 & 1,3 \tabularnewline \hline
1,1 & 3,1 & -1,2 \tabularnewline \hline
0,1 & 1,3 & 1,0 \tabularnewline \hline
\end{tabular}
\end{center}
```

dot나 comma를 기준으로 정리할 때는 컬럼지시자 D를 이용하는데, 다음과 같이 컬럼지시자를 재정의하여 쓰면 간편하다.

컬럼지시자 D

```
\newcolumntype{d}[1]{D{,}{,}{#1}}

\begin{center}
\begin{tabular}{|d{-1}|d{-1}|d{-1}|} \hline
0,1 & 1,-3 & 1,3 \tabularnewline \hline
1,1 & 3,1 & -1,2 \tabularnewline \hline
0,1 & 1,3 & 1,0 \tabularnewline \hline
\end{tabular}
\end{center}
```

0,1	1, - 3	1,3
1,1	3,1	-1,2
0,1	1,3	1,0

6.2 array 환경

표준 array 환경에서 확장된 기능은 left와 right의 괄호 쌍을 편리하게 나타낼 수 있다는 것이다. 여는 괄호와 닫는 괄호는 같은 모양이어야 한다. 한 쪽 괄호만 쓰고자 할 때는 period(.)와 함께 짝을 이루도록 하면 된다.

$$\left(\begin{array}{cc|c} x_1 & x_2 & x_{12} \\ x_3 & x_4 & \\ \hline & x_{21} & x_{22} \\ & x_{31} & x_{32} \end{array} \right)$$

```
\[
\begin{array}{|cc|}
\begin{array}{cc} x_1 & x_2 \\ x_3 & x_4 \end{array} & x_{12} \\
\hline
& x_{21} & x_{22} \\
& x_{31} & x_{32}
\end{array}
\]
```

주의할 것은 대괄호 [와]는 position 옵션과 함께 사용해야 한다. 그렇지 않으면 [가 여는 괄호인지 옵션의 시작인지를 구분할 수 없기 때문인 것으로 보인다.

$$\left[\begin{array}{cc} x_{11} & x_{12} \\ x_{21} & x_{22} \end{array} \right]$$

```
\[
\begin{array}[c]{cc}
x_{11} & x_{12} \\
x_{21} & x_{22}
\end{array}
\]
```

6.2.1 표(Tables)

6.2.1.1 toprule, bottomrule, midrule, cmidrule

표에서는 수직패션을 굵지 않는 것을 권장한다. 표준 L^AT_EX에서와는 달리, memoir에서는 수직패선의 굵기 만큼 표의 가로길이가 늘어난다.

표 6.1: Do not use vertical lines

Animal	Description	Price (\$)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

`\toprule`
`\midrule`
`\bottomrule`
`\cmidrule{n-m}`

```

\begin{table}[htb]
\centering
\caption{Do not use vertical lines}
\label{tab:noververticallines}
\begin{tabular}{@{}llr@{}}
\toprule
\multicolumn{2}{c}{Item} \\
\midrule
Animal & Description & Price (\$) \\
\midrule
Gnat & per gram & 13.65 \\
& each & 0.01 \\
Gnu & stuffed & 92.50 \\
Emu & stuffed & 33.33 \\
Armadillo & frozen & 8.99 \\
\bottomrule
\end{tabular}
\end{table}

\cmidrule[<wd>](<trim>){<n-m>}

```

`\cmidrule`은 n열부터 m열까지 midrule을 connect해준다. trimming을 위해서는 (r), (l), (lr)이 오는데, 예를 들면, (r)은 패선의 오른쪽을 trimming 하라는 표시이다. 혹시라도 필요하다면 [.8pt]와 같이 패선의 두께를 바꿀 수 있다. 패선의 두께를 바꾸는 방법은 top, mid, bottomrule 경우에도 모두 같다.

그럴리 없겠지만, 혹시라도 두 개의 패선이 필요하다거든 다음과 같이 하면 된다.

```
\toprule \toprule  
\cmidrule(lr){1-2}\morecmidrules\cmidrule{1-2}
```

행과 행사이에 2mm의 공백을 추가하고 싶으면, \\ 다음에 `\addlinespace[2mm]` 를 `\addlinespace` 넣으면 된다. 이 것은 2mm의 투명한 패션을 긋는 것이라고 생각하면 된다.

6.2.1.2 tabular 환경

`tabular*` 환경은 컬럼 사이의 공간을 변경함으로써 테이블 폭을 조절한다. `tabularx` 환경은 컬럼의 폭을 변경하여 테이블 폭을 변화시킨다.

다음에 공부하자.

제 7 장

newenvironment 연습

7.1 one example: environment

다음 소스는 새로운 환경을 정의하고, 정의된 새로운 환경을 사용하는 예이다.

```
\newcounter{EXAMP}[chapter]\newsavebox{\EXAMP}
\makeatletter
\newenvironment{emps}
{\stepcounter{EXAMP}%
\renewcommand{\theEXAMP}{\thechapter.\arabic{EXAMP}}%
\def\@currentlabel{\p@EXAMP\theEXAMP}%
\noindent \hrulefill\ \textbf{예: \theEXAMP}}
{\noindent \hrulefill}
\makeatother

\begin{emps}\label{emps:first}
어찌구 저찌구 ...
\end{emps}
```

그 결과는 다음과 같다.

예: 7.1 어찌구 저찌구 ...

위의 정의에서 `\p@EXAMP`는 `\pageref`를 위한 정보이고 `\theEXAMP`는 `\ref` 명령을 위한 정보이다. L^AT_EX은 컴파일 도중 이 정보들을 `foo.aux` 파일에 기록한다.

그렇게 하면, (`\ref{emps:first}`에 의해) 7.1 또는 (`\pageref{emps:first}`에 의해) 41 페이지와 같이 번호를 참조할 수 있다.

7.2 another example: environment

```
\newenvironment{exer}{\noindent\hrulefill\ \textbf{연습문제: }}%
  {\noindent \hrulefill}
\newenvironment{exerans}{\noindent\hrulefill\ \textbf{연습문제답: }}%
  {\par\noindent \hrulefill}
```

위와 같이 ‘연습문제’와 ‘연습문제답’을 위한 환경을 정의하고 이를 다음과 같이 사용한다.

```
\begin{exer}
다음 문제를 풀어라.

. . .

\end{exer}
```

연습문제: 다음 문제를 풀어라.

1. $f(x) = e^x$ 의 $x = 0$ 에서의 선형근사를 구하라.
2. 차원이 같은 정사각행렬 A, B 에 대하여, $AB = BA$ 가 사실인지의 여부를 밝혀라.

문제를 쳤으니, 이제 답을 치자. 물론 답을 위한 환경은 다음과 같이 이용한다.

```
\begin{exerans}
그런데, 답은 여기에 없다.
\end{exerans}
```

연습문제답: 그런데, 답은 여기에 없다.

参考文献

`\prebibhook`: 이 참고문헌은 테스트를 위해서 작성한 것이므로, 실제 문서의 내용과 관련이 없다. 참고문헌 항목이 나열되기 전에 이 문장을 식자하기 위해 `\prebibhook` 을 이용하였다.

김강수 옮김, 『The LaTeX Memoir 매뉴얼』, *The Memoir Class for Configurable Typesetting User Guide*, by Petter Wilson, 2004.

김강수, 『LaTeX-Memoir로 책만들기』, 2006.

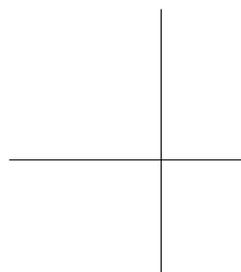
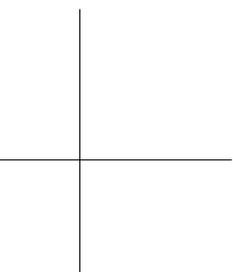
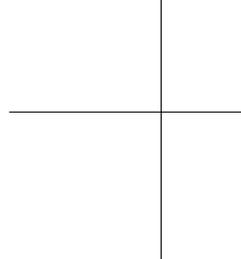
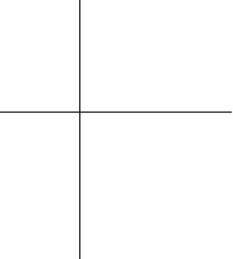
Haeussler, Ernest F., Richard S. Paul, and Richard Wood, *Introductory Mathematical Analysis*, 11th ed., 2003.

Hughes-Hallett, Deborah, Andrew M. Gleason, and William G. McCallum, *Calculus: Single and Multivariable*, 4th ed., 2005.

Stewart, James, *Calculus*, 5th ed., 2005.

Sydsaeter, Knut, and Peter Hammond, *Essential Mathematics for Economic Analysis*, 2002.

`\postbibhook`: CTAN is the Comprehensive TeX Archive Network and URLs for the several CTAN mirrors can be found at <http://tug.org>. `\postbibhook` is used to place this sentence after all the bibitems. Note that `\postbibhook` command must be typed before `\begin{thebibliography}`.



찾아보기

\prebibhook: 찾아보기가 나오기 전에 쓸 말이 있으면 여기에 쓴다.

【 M 】

makeindex.exe 24

【 ㅈ 】

정리 24

 재고정리 24

 점포정리 25

 정점정리 24