

# README

## pgreenbook

pgreenbook 패키지는 2013년 공주대학교 문서 작성 워크숍에서 이주호(Progress)가 강연한 "견고하게 작성된 텍 원고를 출판에 적합하게 고쳐보기" 주제의 발표에서 실습 과제로 제시되었던 단행본 디자인 샘플이다.

그 후 여러 사람이 이 샘플을 이용하여 책을 제작하거나 문서를 만들게 되었고, 이를 KTUG Wiki의 [DocumentGallery] 페이지에서는 초록책이라고 부른다.

이주호와 노바 데 히가 현재의  $\text{\LaTeX}$  작업 환경에서도 무리없이 컴파일되도록 코드를 조금씩 수정해가고 있다.

처음에 스타일의 이름이 KNUworkshop2013.sty이었으나 이를 독립된 패키지로서 릴리스하면서 pgreenbook으로 바꾸었다. *p*는 Progress의 첫 글자이다.

## pgreenbook sample

패키지와 함께 제공되는 샘플 문서는 oblivoir로 작성하는 단행본이다. 텍스트는 남상호, 《경제변동론》과 Knuth의 글 등 이곳저곳에서 가져왔다.

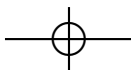
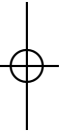
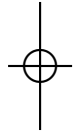
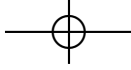
이 예제 문서를 컴파일하여 봄으로써 패키지가 의도하는  $\text{\LaTeX}$  문서 디자인을 더 잘 이해할 수 있을 것이다.

문서는 pgreenbooksample.zip 파일을 다운로드받아서 이용할 수 있으며 패키지가 설치된 texmf-dist/doc/latex/pgreenbook/ 폴더에서도 찾을 수 있다.

# 차 례

README	i
차 례	ii
제 1 장 성마른 독자를 위한 요약	1
제 2 장 책의 기본 요소	2
2.1 판형과 재단선	2
2.2 본문 줄 간격	3
2.3 폰트	4
2.4 색상	5
2.5 memoir와 tikz	6
제 3 장 장(chapter)과 페이지	7
3.1 chapter style	7
3.2 페이지 스타일	9
3.2.1 페이지 스타일 기초	9
3.2.2 running heading의 장절표제 텍스트	10
3.2.3 몇 가지 보충	13
3.3 section 표제 스타일	13
제 4 장 목차와 캡션	14
4.1 Table of Contents의 이해	14
4.1.1 길이	14
4.1.2 폰트	15
4.1.3 <num> 파트	15
4.1.4 <title> 파트	17
4.1.5 <leaders> 파트	17

4.1.6	<pnum> 파트 . . . . .	17
4.1.7	페이지 번호 이후 . . . . .	18
4.2	이 책의 TOC . . . . .	18
4.3	캡션 . . . . .	19
제 5 장	인덱스 . . . . .	20
5.1	자동 인덱스 명령 . . . . .	20
5.2	printindex 스타일링 . . . . .	20
제 6 장	정리류 . . . . .	22
6.1	정리, 명제, ... . . . .	22
6.2	박스 디자인에 대한 코멘트 . . . . .	23
찾아보기		24



## CHAPTER 1

## 성마른 독자를 위한 요약

이 스타일을 이용하여 책을 만들려면,

(1) KoPub World 서체를 설치하고 사용 가능하게 한다.

(2) Preamble을 다음과 같이 작성한다.<sup>1</sup>

```
\documentclass[showtrims,chapter]{oblivoir}  
\usepackage{pgreenbook}
```

(3) 문서 시작부분(`\begin{document}` 이후)에 다음과 같이 써준다.

```
\pagestyle{KNUworkshop}  
\chapterstyle{KNUchapter}
```

<sup>1</sup>[chapter] 옵션을 선언하면 `openright`, `twoside`가 기본이 된다.

## CHAPTER 2

## 책의 기본 요소

이하 pgreenbook으로 만든 책을 “이 책”이라 한다.

## 2.1 판형과 재단선

```
\usepackage[stock]{fapapersize}  
\usefastocksize{210mm,297mm}  
\setheadfoot{\headheight}{12mm}  
\usefapapersize{174mm,251mm,27mm,*,35.5mm,*}  
  
\quarkmarks
```

‘크라운판’이라 불리는 사이즈는 보통 175×248mm (18절)로 재단한다. 이 책의 사이즈는 크라운판에 가까운 변형판본이다. 인쇄용지를 A4로 보고 재단선을 긋고 있다.

재단선 명령은 `\quarkmarks`이다.

이 디자인에 의하면 글줄 길이(`\textwidth`)가 120mm 가량이 된다. 실제로 그려한지 짚어보자.

<code>\textwidth</code>	119.84842mm
<code>\spinemargin</code>	27.00014mm
<code>\foremargin</code>	27.00014mm

이 파라미터들은 모두 memoir에 설명이 자세히 나와 있다. 만약 마진 문단을 디자인할 생각이라면 좀더 세심한 주의가 필요하다. `\marginparwidth` 및

`\sideparwidth` 등이 항상 `\spinemargin` 또는 `\foremargin`과 일치하지 않기 때문이다.

## 2.2 본문 줄 간격

```
\SetHangulspace{1.4}{1.12}
\RequirePackage[mathleading=1.2]{ob-mathleading}
```

`\SetHangulspace`는 **oblivoir** 명령이다. 두 개의 인자를 취하며 앞의 것이 본문 행송, 뒤의 것이 ‘좁은 행송’인데 이것은 특히 footnote와 floats 안의 간격을 제어한다. 자세한 사례는 **oblivoir-test** 문서를 참고하라.

행송은 `\linespread`값으로서 달리 `\baselinestretch`라고 부르는 값이다. **oblivoir** 자체 기본값은 1.333인데, 이 책에서는 1.4로 하고 있다. 이것이 무엇을 의미하는지 계산을 조금 해보자.

본문 10pt 문서에서  $\text{\TeX}$ 은 12pt를 행간격 기본값으로 설정한다. 이것은 원래 폰트 속성으로서 예컨대 `\fontsize`명령의 두 번째 인자와 같다. 즉,

```
\fontsize{12pt}{14pt}
```

이 명령은 글자를 12포인트로, 윗줄과 아랫줄 **baseline**의 간격을 14포인트로 설정한다는 의미이다.

본문 10포인트의 경우, 즉 행 사이 12포인트를 고려할 때, 1.33이라는 값은  $12 \times 1.333 = 15.996$ 을 의미하게 되는데, 이것은 HWP의 기본 행간인 160%에 매우 근사한 값이다. 즉 **oblivoir**는 행간이 HWP와 매우 비슷하게 보이도록 디자인되어 있는 것이다. 이 책의 1.4는 16.8포인트로서, 168%에 상당하는 값이다. 이 정도 간격이 한글 책 독자들에게 잘 받아들여진다고 한다.

소위 배행간이라 불리는 200%는 얼마나 값을 주어야 할까? 1.67 정도가 거의 정확하게 200% 행간격을 설정하게 한다.

**ob-mathleading** 패키지는 **amsmath**-류 여러 줄 수식의 행 사이 간격을 제어하는 패키지이다. 본문보다는 좁은 행간을 지정해야 보기 좋은 수식이 나온다.

```
texdoc ob-mathleading
```

명령으로 설명서를 읽어보아라.

## 2.3 폰트

```
\setmainfont{STIX Two Text}
\setsansfont{Source Sans Pro}
  [Scale=MatchUppercase,BoldFont={* Semibold}]
\setkomainfont[KoPubWorldBatang ](Light)(Medium)
\setkosansfont[KoPubWorldDotum ](Light)(Medium)
\ifXeTeX
\newfontfamily\fallbackhanjafont{Noto Serif KR}
  [Scale=.92]
\etextkofontregime[puncts=prevfont, colons=prevfont,
  cjksymbols=hangul]{latin}
\fi

\usepackage{unicode-math}
\setmathfont{latinmodern-math.otf}
\setmathhangulfont{KoPubWorldBatang Light}
```

폰트 설정 부분이다. 다음과 같이 설정되어 있다.

라틴 문자 메인	STIX Two Text
라틴 문자 산세리프	Source Sans Pro
한글/한자 메인	KoPub World Batang
한글/한자 산세리프	KoPub World Dotum
수학 기본	Latin Modern Math
수학 한글	KoPub World Batang

이 스타일에서는 `\etextkofontregime`으로 문장부호의 식자 폰트를 지정했는데, 이 명령은  $\text{XeTeX-ko}$ 에서만 쓸 수 있다. 대체로 이 설정이 없어도 큰 위화감 없이 잘 나온다. 자세한 설명은  $\text{XeTeX-ko}$  매뉴얼에 나오므로 이를 숙지한 후에 디자인 의도를 구현하는 명령을 작성하라.

수학식 내에서 한글을 `\text` 명령 없이 쓰려면 `\setmathhangulfont`를 절정해두어야 한다. `unicode-math`를 쓰지 않는 경우에도 그러하다.

이 패키지에 옵션으로 `[exthanja=true]`를 주면 `fallback` 한자에 Noto Serif KR 폰트를 적용한다. 기본 한자는 한글 폰트의 것을 사용한다. 이 옵션은 복잡한 한자가 필요한 경우에 쓸 것이다. Noto Serif CJK KR이 아니라 Noto Serif KR임에 주의하라. 물론 폰트는 별도로 설치되어 있어야 한다.











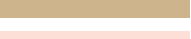






## 2.4 색상

```
%% 색깔설정
\@ifpackageloaded{xcolor}{}{
  \RequirePackage[dvipsnames,svgnames,x11names]
    {xcolor}
}
% Pantone Fashion Color: Spring 2014
\definecolor{PlacidBlue}{cmyk}{.47,.17,.02,.0}
\definecolor{VioletTulip}{cmyk}{.44,.39,.0,.0}
\definecolor{Hemlock}{cmyk}{.39,.04,.35,.0}
\definecolor{Paloma}{cmyk}{.35,.24,.27,.0}
\definecolor{Sand}{cmyk}{.20,.27,.48,.0}
\definecolor{Freesia}{cmyk}{.0,.14,.100,.0}
\definecolor{Cayenne}{cmyk}{.06,.74,.56,.0}
\definecolor{CelosiaOrange}{cmyk}{.0,.63,.80,.0}
\definecolor{RadiantOrchid}{cmyk}{.32,.65,.0,.0}
\definecolor{DazzlingBlue}{cmyk}{.92,.57,.0,.0}
\definecolor{PurpleHaze}{cmyk}{.56,.51,.15,.0}
\definecolor{Comfrey}{cmyk}{.74,.28,.63,.10}
\definecolor{MagentaPurple}{cmyk}{.51,.94,.24,.24}

\colorlet{MainColorOne}{Comfrey}
\colorlet{MainColorTwo}{PurpleHaze}
```

단행본 디자인에서 색상은 매우 중요하다. 인쇄물을 목적으로 하는 때와 온라인 출판을 의도하는 때에 사용할 수 있는 색공간이 다르기 때문이다. 인쇄물에 (변환되지 않은) RGB 값을 쓰면 색공간이 어그러지기 때문에 특별한 주의가 필요하다.

이 책에서는 모두 cmyk 색공간으로 사용할 색을 지정하고 있다. 화면에서 보던 익숙한 색과 색감에 차이를 보이는 이유는 색공간이 cmyk이기 때문이다.

PlacidBlue		RadiantOrchid	
VioletTulip		DazzlingBlue	
Hemlock		PurpleHaze	
Paloma		Comfrey	
Sand		MagentaPurple	
Freesia		MainColorOne	
Cayenne		MainColorTwo	
CelosiaOrange			

이 책이 '초록책'이라 불리는 이유는 주판면에 사용한 색상 MainColorOne

## 제2장 책의 기본 요소

때문일 것이다.

만약 2도 인쇄 또는 별색(spotcolor)을 사용하는 책을 디자인한다면 거기에 맞춘 특별한 설계와 주의가 필요하나, 이 글에서 다룰 수는 없다. 이 책이 4도 인쇄를 위하여 디자인되었음을 지적하는 것으로 그친다.

## 2.5 memoir와 tikz

```
\RequirePackage{tikz}
\RequirePackage{tikzpagenodes}
%\RequirePackage{memtikzpagenodes}
%\RequirePackage{oblivoir-misc}
```

TikZ를 페이지 디자인에 활용하려 할 때, tikzpagenodes 패키지는 매우 요긴하다. 그런데 이 패키지를 그대로 쓸 때는 이것이 memoir의 페이지 요소와 일부 어긋날 수 있다는 것을 알아야 한다.

Kriss가 기여한 바, memoir+tikzpagenodes 호환을 위한 삼절기라는 글에 이 문제를 다루고 있고, memtikzpagenodes.sty를 제시하고 있다.

한편 tikzpagenodes를 쓰지 않고 단지 TikZ의 current page 노드만을 쓴다 해도 역시 비슷한 일이 일어난다. 이 문제는 oblivoir-misc 패키지를 로드하는 것으로 해결된다.

다만, 이 책에서는 그러한 “어긋남”을 전제로 페이지 디자인을 행했다. 즉 어차피 어긋난다고 생각하고 그것을 수치 보정하여 해결한 것이다. 따라서 이 스타일을 쓸 적에는 위에 언급한 두 패키지를 굳이 로드할 필요는 없다. (memtikzpagenodes를 로드하면 이 책의 원래 디자인과 달라질 수 있다.)

## CHAPTER 3

## 장(chapter)과 페이지

## 3.1 chapter style

```
\makechapterstyle{KNUchapter}{%
  \setlength{\afterchapskip}{40pt}
  \renewcommand*{\chapterheadstart}{
    {\vspace*{-3\onelineskip}}
  }
  \renewcommand*{\afterchapternum}{
    {\par\nobreak\vskip 25pt}
  }
  \renewcommand*{\chapnamefont}{
    {\normalfont\LARGE\flushright}
  }
  \renewcommand*{\chapnumfont}{
    {\normalfont\HUGE\color{MainColorOne}}
  }
  \renewcommand*{\chapttitlefont}{
    {\color{MainColorOne}\sffamily\HUGE\bfseries\flushright}
  }
  \renewcommand*{\prechapternum}{
    {\chapnamefont\MakeUppercase{Chapter}}
  }
  \renewcommand*{\postchapternum}{%
    \makebox[0pt][l]{%
      \hspace{1em}{\color{MainColorOne!50}%
        \rule{\midchapskip+\spinemargin}{\beforechapskip}}
    }
  }
  \renewcommand*{\chapternamenum}{%
    \setlength{\beforechapskip}{18mm}% \numberheight
    \setlength{\midchapskip}{\paperwidth}% \barlength
    \addtolength{\midchapskip}{-\textwidth}
    \addtolength{\midchapskip}{-\spinemargin}
    \renewcommand*{\printchapternum}{%
      \hspace{1em}\resizebox{!}{\beforechapskip}%
      {\chapnumfont \thechapter}}
  }
  \makeoddfoot{plain}{}{}
  {\normalfont\normalsize\sffamily\thepage}
}
```

## 제3장 장(CHAPTER)과 페이지

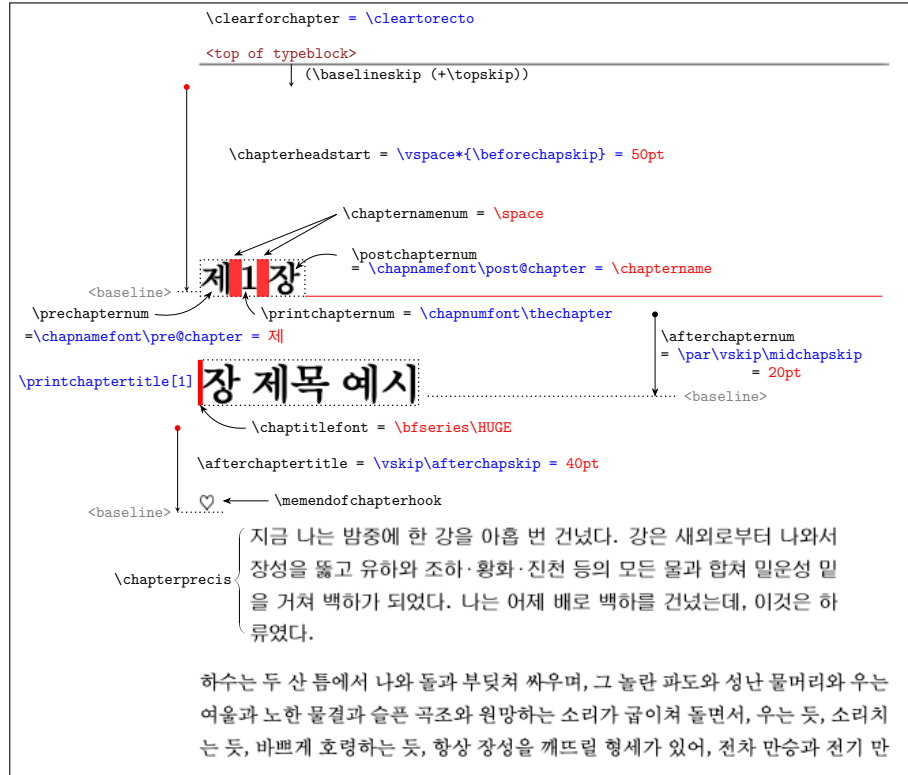


그림 3.1 oblivoir의 장 표제 파라미터

이 책은 KNUchapter라는 이름의 장 스타일을 정의한다. 장 스타일의 정의는 기본적으로 장을 이루는 각 파라미터 매크로를 재정의하는 것이다. 각 파라미터들을 설명하는 것은 oblivoir 매뉴얼에 나오는 그림 3.1로 충분히 설명이 될 것이다.

이 그림에 나오는 각 파라미터들을 어떤 식으로 재정의하여 이 책의 장 스타일을 만들었는지 잘 살펴보는 것이 공부가 될 것이다. 필요하다면 tikz를 활용하여 여러 도형이나 선을 긋거나 그림 등을 활용하여 꾸밀 수 있다. 어느 위치의 매크로를 재정의하는 것이 가장 좋을지는 그때그때 상황에 따라 다르므로 약간의 아이디어가 필요한 경우도 있다.

한편 한글 서적에서는—이 책은 그러하지 않지만—장 표제면이라 하여 장 표제를 하나의 독립 페이지로 꾸미는 일도 많다. 이 경우 가장 간단한 방법은 \memendofchapterhook에 \clearpage 또는 \cleartorecto를 넣어주

## 3.2 페이지 스타일

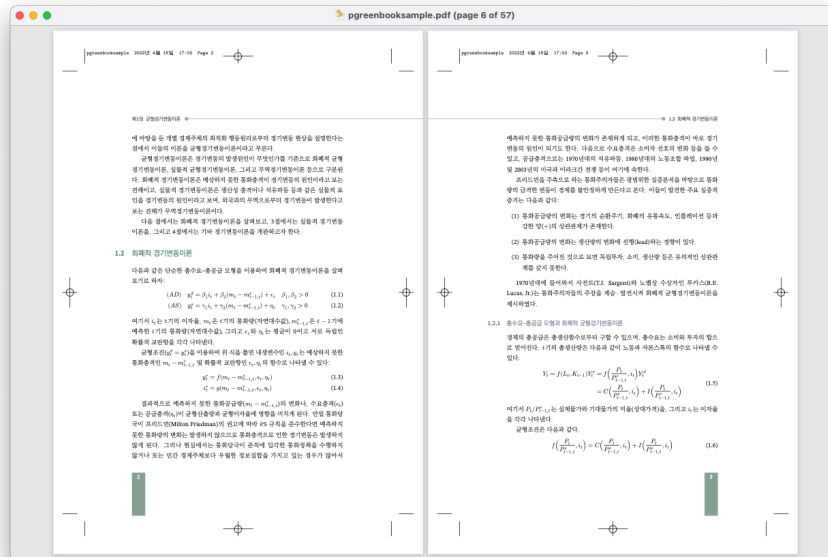


그림 3.2 KNUworkshop 페이지 스타일

는 것이다.

### 3.2 페이지 스타일

이 책은 두 종류의 페이지 스타일을 정의한다. 각각 KNUworkshop과 PageStylePrinciple이라는 이름으로 정의되었다. 시각적인 모양은 각각 그림 3.2와 그림 3.3에 나타나 있다.

#### 3.2.1 페이지 스타일 기초

pagestyle을 정의하는 것은 어렵지 않다. 먼저 페이지스타일의 이름을 짓는다.

```
\makepagestyle{<pagestyle>}
```

하나의 페이지의 상단(head)과 하단(foot)에 각각 left, center, right로 배열되는 세 개의 ‘영역’이 있다고 생각하자. 그러면 모두 여섯 개의 ‘영역’에 무언가를 넣을 수 있다. 양면 조판의 경우 좌우 페이지가 달라지므로 12종류의 ‘뭔가 넣을 것’을 생각할 수 있고, 이것을 다음 매크로로 정의해주는 것이다.

## 제3장 장(CHAPTER)과 페이지

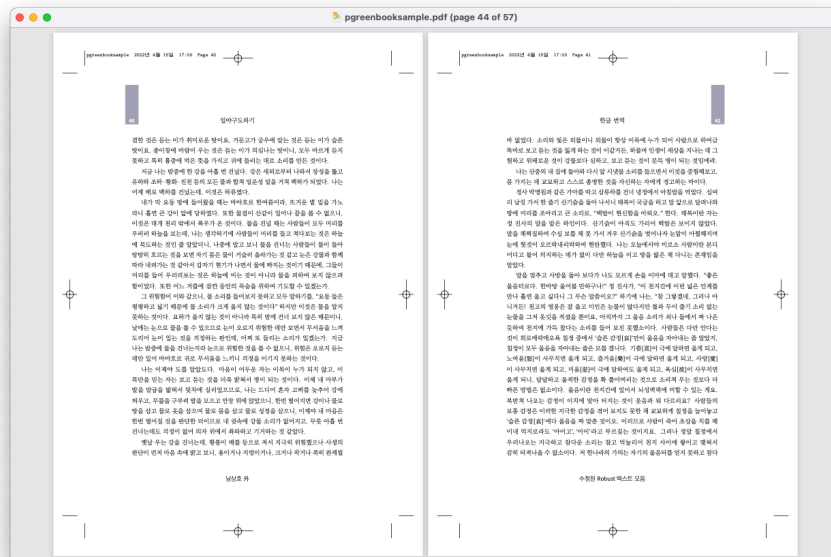


그림 3.3 PageStylePrinciple 페이지 스타일

```
\makeoddhead {\pagestyle}{\left}{\center}{\right}
\makeoddfoot {\pagestyle}{\left}{\center}{\right}
\makeevenhead{\pagestyle}{\left}{\center}{\right}
\makeevenfoot{\pagestyle}{\left}{\center}{\right}
```

‘뭔가 넣을 것’은 뭐라도 좋다. 텍스트, 박스, tikz 드로잉 등이 주로 오는 것들이다.

추가로 고려할 것은 \makerunningwidth이다. 페이지의 \textwidth와 다른 길이를 가지도록 할 수 있어서 판면보다 넓은 헤딩을 작성할 수 있게 한다.

페이지 스타일 정의 부분의 소스 코드는 인용하지 않겠다, 너무 길어서. 스타일 파일의 해당 부분을 잘 공부해 보면 얻는 것이 있을 것이다.

## 3.2.2 running heading의 장절표제 텍스트

TEX에는 \leftmark와 \rightmark라는 중요한 매크로가 있다. 원래 아무 것도 들어 있지 않은데, \markboth와 \markright라는 명령에 의해 그 내용이

채워진다.

`\markboth` 두 개의 인자를 취하여, 앞의 것을 `\leftmark`에, 뒤의 것을 `\rightmark`에 할당한다.

`\markright` 한 개의 인자를 취하여 그것을 `\rightmark`에 할당한다.

중요한 것, `\markleft`라는 명령은 없다는 사실. `\markleft`를 하고 싶다면 먼저 `\markboth`를 실행하고 `\markright`로 빈 인자(또는 다른 것)를 차례로 주면 된다.

이 두 명령은 문서 중에 사용하는 것이 불가능하지는 않지만, 표준  $\text{\LaTeX}$ 에서는 `\chaptermark`나 `\sectionmark` 같은 다른 명령에 의해 불리도록 설계되어 있다. 즉, `\chapter` 명령은 그 자체가 `\chaptermark`를 부르게 되는데 그 내용은 `\markboth`를 실행하는 것이다. 한편 `\section` 명령은 그 자체가 `\sectionmark`를 부르게 되는데 그 내용은 `\markright`를 실행하는 것이다. 그 결과, `\chapter`와 `\section`이 차례로 실행되고 나면 chapter heading 타이틀은 `\leftmark` 매크로에, section heading 타이틀은 `\rightmark`에 들어가 있게 된다. 굳이 “heading 타이틀”이라고 한 이유는 장 표제와 heading 타이틀이 동일하지 않을 수도 있기 때문이다. 예컨대 `\chapter[heading 타이틀]{장 표제}`와 같은 명령을 부여하면 장 표제로 찍히는 것은 “장 표제”라는 문자열이지만 heading 타이틀은 “heading 타이틀”이라는 문자열이 된다. 이 때 `\leftmark`에 들어 있는 내용은 “heading 타이틀”이다.

`memoir/oblivoir`에는 굉장히 중요한 특징이 한 가지 있다. 즉 `pagestyle`마다 `running heading`을 다르게 찍히도록 할 수 있는 것이다. 그것은 `\makepsmarks`라는 것인데, 기본 명령꼴은 다음과 같다.

```
\makepsmarks{<pagestyle>}{<running-headings>}
```

그리고 이 명령의 두번째 인자 자리에서 사용할 `\createmark`라는 명령이 있다. `memoir` 매뉴얼에 이에 대한 자세한 설명을 꼭 한 번 읽어보기 바란다. 여기서는 이 책에서 사용하고 있는 범위에서 다음과 같은 사용법만을 소개하고 지나간다.

```
%% psmarks
\makepsmarks{KNUworkshop}{%
  \createmark{chapter}{left}{shownumber}
  {\pre@chapter}{\post@chapter\enskip}
  \createmark{section}{right}{shownumber}{\enskip}
```

```

}
\makepsmarks{PageStylePrinciple}{%
  \createmark{chapter}{left}{nonumber}{}{}
  \createmark{section}{right}{nonumber}{}{}
}

```

\createmark는 무려 다섯개의 인자를 취한다. 순서대로 (1) 문서 섹셔닝 레벨 (2) mark 위치 (3) 번호 형식 (4) 번호 앞에 오는 것 (5) 번호 뒤에 오는 것.

- (1) 문서 섹셔닝 레벨. chapter, section, subsection 등
- (2) mark 위치. left, right, both 셋 중의 하나.
- (3) 번호 형식. shownumber, nonumber, notitle 셋 중의 하나. 이 가운데 notitle은 “2.3절 절제목”과 같은 러닝 헤딩 중에서 “2.3절”에 해당하는 부분만 표시한다는 뜻이다.
- (4) 번호의 앞부분. 예를 들면 chapter 1에서 “chapter”에 해당하는 것이다.
- (5) 번호의 뒷부분. 예를 들면 chapter 1과 chapter title 사이에 무엇을 둘 것인가, 즉 장번호의 숫자를 찍은 뒤에 둘 것이다.

nonumber의 경우에 4,5번 인자는 단순히 빈 인자 {}로 둔다.  
단 한 줄만 보자.

```

\createmark{chapter}{left}{shownumber}
  {\pre@chapter}{\post@chapter\enskip}

```

이것은 KNUworkshop 페이지 스타일에서, \chapter 명령이 불리면

```

\leftmark =
  \pre@chapter \value{chapter}\post@chapter
  \enskip <titletext>

```

이 되도록 하라는 뜻이 된다. 지금 이 위치에서

제3장 장(CHAPTER)과 페이지

이것이 그 결과이고 실제로 짝수쪽 헤딩에 이렇게 나타나고 있을 것이다.



### 3.2.3 몇 가지 보충

- hangul 페이지 스타일은 oblivoir에서 쓸 수 있는 한국식 페이지 스타일이다.
- `\coppagestyle`은 이미 있는 스타일을 복사하여 수정할 수 있게 한다.
- `\aliaspagestyle`은 특별한 목적의 페이지 스타일을 이미 있는 스타일과 같게 만들 수 있다.

중요한 것은 `chapter`라는 이름의 페이지 스타일인데, `\chapter` 명령이 불리는 단 한 페이지는 이 스타일이 된다. 기본값이 `plain`이므로 반드시 원하는 모양으로 `alias`해주어야 한다. 이 책에서는

```
\coppagestyle{chapter}{KNUworkshop}
\makeoddhead{chapter}{}{}{}
\makeevenhead{chapter}{}{}{}
```

과 같이 하여 `\chapter`가 쓰이는 페이지의 모양을 바로잡아 주고 있다.

## 3.3 section 표제 스타일

```
\setsechhook{%
  \setsecnumformat{\thesection\quad}
  \setsechheadstyle{\sffamily\bfseries\Large\color{MainColorOne}}
  \setsubsechheadstyle{\sffamily\bfseries\color{MainColorTwo}}
  \hangsecnum
}
```

`section`보다 낮은 수준의 문서 구분 명령의 표제 스타일은 `\setsechheadstyle`이나 `\setsechhook`, `\secnumformat` 등을 사용하여 장식할 수 있다.

이 책에서는 위와 같이 선언하였다. `\hangsecnum`은 절의 번호가 판면 밖으로 나오도록 설정하는 것이다.

# CHAPTER 4

## 목차와 캡션

### 4.1 Table of Contents의 이해

이 책의 TOC 디자인에 대해 알아보기 전에 memoir/oblivoir가 이 부분을 어떻게 만드는지 대략 살펴보자. 그림 4.1에 표시된 각 부분별로 설명하고자 한다.

제 1 장 jiwonlinsum 패키지	1
1.1 한글 부분	1
1.1.1 한글 부분의 구성	4
1.1.2 문단 표지가 없는 경우	4
1.1.3 요약	5
1.2 한자 부분	5
1.2.1 확장 한자의 문제	6

그림 4.1 oblivoir의 toc

#### 4.1.1 길이

그림 4.1에 표시된 길이 변수는 다음 세 가지가 있다: `\cftKindent`, `\cftKnumwidth`, `\@pnumwidth`. 이 가운데 마지막의 `\@pnumwidth`는  $K(=chapter, section, etc)$ 에 따라 달라지지 않고 일정하기 때문에 이를 바꾸기 위해서는

```
\setpnumwidth{<dim>}
```

으로 설정한다. 그림에는 나와 있지 않지만 페이지 번호가 찍히는 위치가 typeblock의 오른쪽 끝이 아니어야 할 때는 `\@tocrmarg`라는 길이 변수를 조절할 수 있다. 이름 그대로 오른쪽 마진을 추가하는 것이다. 이를 설정하려면

```
\setrmarg{<dim>}
```

이 값은 기본적으로 0pt이다.

chapter, section 등의 수준에 따라서 indent와 numwidth를 별도로 지정할 수 있는데 이들은 길이값이므로 `\setlength`로 변경해도 되지만

```
\cftsetindents{<kind>}{<indent>}{<numwidth>}
```

명령을 쓰는 것이 좋다. `\cftsetindents{chapter}{0pt}{3.5em}` 이렇게 하면 `\cftchapterindent 0pt`, `\cftchapternumwidth 3.5em`으로 설정한다.

#### 4.1.2 폰트

`\cftchapterfont`, `\cftsectionfont` 등의 매크로는 그 행 전체의 폰트를 지정하는 명령이 된다. 앞으로 설명할 것과 같이 원한다면 각 세부 부분의 폰트를 별도로 지정할 수 있지만 그렇게 하지 않으면 이 매크로가 페이지 번호까지 —`\cftKpagefont`를 설정하지 않았다면— 영향을 미친다. `\renewcommand`로 바꿀 수 있다.

#### 4.1.3 <num> 파트

그림에서 (1)로 표시된 부분을 편의상 “<num> 파트”라고 부르기로 하자. 즉 chapter에서는 “제 1 장”, section에서는 “1.1”에 해당하는 부분이다.

이 부분을 식자하라는 memoir 내부 명령이 `\chapternumberline`과 `\numberline`이다. 달리 말하면 chapter 레벨의 `\numberline`은 `\chapternumberline`이다. section 이하 레벨에서 `\numberline`은 그대로 `\numberline`을 사용한다. 이 명령들은 그 인자로 식자할 <num> 파트의 내용(주로 번호)를 받아서 이를 다음과 같이 처리한다.

- (i) 먼저 `\chapternumberlinehook` 또는 `\numberlinehook`이 정의되어 있다면 이를 실행한다. 당연히 디폴트는 아무 것도 하지 않는 것이다.

(ii) `\chapternumberlinebox` 또는 `\numberlinebox`라는 박스를 만들고 이를 식자한다. 이 박스 명령은 두 개의 인자를 취하여, 하나는 박스의 가로 길이이고 다른 하나는 `\numberline`에서 넘겨받은 “식자할 것”이다.

여기 설명한 매카니즘을 이용하여 `<num>` 파트 식자 방법을 원하는 대로 바꿀 수 있다. 그러나 그런 것은 고급 사용자나 패키지 개발자에게 맡기고, 우리는 디폴트로 주어진 `numberline` 식자 매카니즘에 익숙해지도록 하자. 개략적으로 말하면, `chapter`의 `numberlinebox`의 가로 길이는 당연히 `\cftchapternumwidth`이고, `box`의 구성은 대략 다음과 같다. 다음에서 `K`는 `chapter`와 같은 레벨의 명칭이다.

```
\cftKname \cftKpresnum [NUM] \cftKaftersnum \hfill
```

`\cftchaptername`은 그 이름과 상관없이 일종의 `hook` 명령이고 기본값은 아무 것도 하지 않는 것이다. 이 매크로의 영향은 `<num>` 파트에만 미치므로 이를 활용할 일이 있을 수 있다.

주의: `section` 레벨에서 `\cftsectionpresnum`이 `\hfill`로 되어 있는 것이 기본값임을 주의하라. `section`의 `<num>` 파트의 정렬을 고려할 때 이 점을 잘 기억해야 한다.

`oblivoir`의 `NUM` 영어로 된 책이라면 “Chapter 1”과 같은 `<num>`의 형식에서 “Chapter” 부분은 `\cftchaptername`을 `\chaptername`으로 정의함으로써 구현하고 `[NUM]`에는 그냥 숫자 “1”만 오면 된다.

그런데 한국어 책에서는 이것이 “제 1 장”하는 식으로 숫자의 앞뒤에 뭔가가 붙는다. `oblivoir`는 이를 위해서 `\hchaptertitlehead`라는 매크로를 제공하고 있다. 이것은 아예 `chapter number`에 `\pre@chapter`와 `\post@chapter`를 붙여서 넘겨주는 것이다. (원래 이 매크로는 `toc`와 `page running heading`에서 쓰기 위해 고안된 것이었다.) `\hchaptertitlehead`는 `chapter style`을 정의하면서 `\renewcommand`한다. 그리고 `toc`에는 이것이 찍힌다. 만약 `toc`에서 “제”라는 것을 제거하고 싶다면, `\hchaptertitlehead`를

```
\renewcommand*\hchaptertitlehead{\value{chapter}~\post@chapter}
```

으로 미리 정의해두면 된다. 이것은 본문에서 장 번호를 식자할 때는 영향이 없다.

#### 4.1.4 <title> 파트

그 다음으로 (2)의 <title> 파트가 온다. 이것은

```
\cftKaftersnumb [TITLE]
```

로 식자되는 간단한 것이다. \cftchapteraftersnumb가 [TITLE] 텍스트에 영향을 미치는 매크로임을 알아두는 것으로 충분하다.

그리고 (1)의 <num> 파트와 (2)의 <title> 파트는 다시 하나의 그룹으로 묶여 있다.

```
<num> part <title> part
```

그림의 점선이 이를 표시한다.

#### 4.1.5 <leaders> 파트

(3)의 부분은 보통 점을 찍어서 표현하는 부분으로 <leaders> 파트라고 부르기로 한다.

이 파트를 식자하는 명령이 \cftKleader이다. 기본적으로 다음과 같이 정의되어 있다.

```
\cftchapterleader = \cftchapterdotsep{\cftnodots}
\cftsectionleader = \normalfont\cftdotfill{\cftsectiondotsep}
```

이 매크로들은 모두 \renewcommand한다. 예컨대 section의 점을 좀더 촘촘하게 하고 싶으면 \cftsectiondotsep의 값을 적절하게 줄여본다. 4.5가 기본이고 단위 없이 숫자를 넣어야 한다. chapter에서와 같이 아무 것도 찍지 않으려면 \cftnodots를 이용한다.

#### 4.1.6 <pnum> 파트

(4)는 페이지 번호를 찍는 부분이다. <pnum> 파트라고 부르기로 하자. 이 부분은 하나의 박스인데 그 가로폭은 \@pnumwidth로 정해져 있다. chapter와 section에 따라 다른 폭을 주는 것은 비상식적이므로 그렇게 세분되어 있지 않다.

<pnum> 파트는 \cftKformatpnum로 식자한다. chapter라면 인자로서 페이지 번호를 취하는 \cftchapterformatpnum이라는 명령이 여기서 실행된다.

## 제4장 목차와 캡션

이 매크로는 다시 `\cftchapterformatpnumhook`이라는 명령을 실행한다. 이 명령은 페이지 번호를 인자로 취하여 페이지 번호를 장식하거나 하는 데 쓰일 수 있다. 페이지 번호를 찍을 때 사용하는 폰트는 `\cftKpagefont` 매크로이다. `\renewcommand`할 수 있다.

그러면, 다음처럼 한다면 어떻게 될까?

```
\renewcommand*\cftchapterpagefont{\sffamily\bfseries}
\renewcommand*\cftchapterformatpnumhook[1]{\fbox{#1}}
```

결과를 예측해보자.

## 4.1.7 페이지 번호 이후

한 줄의 목록 행을 식자한 후, 마지막에 오는 것은 다음 두 매크로이다.

```
\cftKafterpnum\par
```

보통은 아무 것도 할 게 없지만, 이 매크로로 할 수 있는 이 중에는 다음과 같은 것이 있다. 즉, `chapter` 목록 행에 대해서, 페이지 번호와 목록 타이틀 사이에 `leaders`를 두지 않고 쉼표와 페이지 번호를 바로 잇대어 붙이고 싶을 때이다.

```
\renewcommand*\cftchapterleader{, }
\renewcommand*\cftchapterafterpnum{\cftparfillskip}
```

여기 쓰인 `\cftparfillskip`은 `\hskip\parfillskip`과 거의 같다.

## 4.2 이 책의 TOC

```
\renewcommand{\cftchapterfont}
{\large\bfseries\color{MainColorTwo}}
\renewcommand{\cftchapternumwidth}{4.0em}

\renewcommand{\cftsectionpresnum}
{\raggedright} % note the double `1'
\renewcommand{\cftsectionnumwidth}{2.3em}
\renewcommand{\cftsectionindent}{4.1em}

\renewcommand{\cftsubsectionindent}{6.5em}
\renewcommand{\cftsubsectionnumwidth}{3.0em}
```

이 책은 본문에서 subsection까지 절번호가 붙고 TOC 목록을 subsection 수준까지 만든다.

```
\maxsecnumdepth{subsection}
\maxtocdepth{subsection}
```

### 4.3 캡션

oblivoir에는 [figtabcapt 옵션이 있어서 한국식 캡션 타이틀을 더 쉽게 만들 수 있게 하고 있다. 이에 대해서는 설명서에 자세하므로 이를 참고하라.

이 책은 이 기능에 의지하지 않고 다음과 같이 간단히 처리하고 있다.

```
\captiondelim{\quad}
\captionnamefont{\small\sffamily}
\captiontitlefont{\small\normalfont}
```

지나가는 길에 한 마디 적자면, 플롯가 아닐 때 캡션을 붙이기 위해 \newfixedcaption 명령으로 새로운 캡션 명령을 만들어 쓰는 것이 가능하다.

```
\newfixedcaption{\mycaptionfig}{figure}
```

이제 \begin{figure} 안에 들어 있지 않은 그림에도 \mycaptionfig 명령으로 캡션을 붙일 수 있다.

만약 캡션을 예컨대

[그림 1] 그림의 캡션

과 같은 모양으로 만들려면 다음과 같이 한다. 일단 [figtabcapt 옵션을 oblivoir에 부여한 후에,

```
\obCaptionFont{\sffamily}
\renewcommand*\obCaptionnameOpen{[}
\renewcommand*\obCaptionnameClose{]}
\captiondelim{\quad}
\captionnamefont{\sffamily}
\captiontitlefont{\normalfont}
```

# CHAPTER 5

## 인덱스

### 5.1 자동 인덱스 명령

찾아보기는 책의 중요한 구성요소지만 원고를 쓰는 입장에서 어떤 단어를 포함할 것인지 결정하고 일일이 `\index` 명령을 달아두는 것이 상당히 피곤하다. 이 책에서는 `\myem`이라는 명령을 정의하고 있다.

```
\myem{우리나라} = 우리나라\index{우리나라}
\myem[우리]{나라} = 나라\index{나라}(우리)\index{우리}
```

이렇게 사용한다.

```
\NewDocumentCommand{\myem}{om}{%
  \IfNoValueTF{#1}%
  {#2\index{#2}}%
  {#2\index{#2} (#1)\index{#1}}}
```

### 5.2 printindex 스타일링

```
\renewenvironment{theindex}{%
  \renewcommand\indexspace{\vskip\onelineskip}
  \@restonecoltrue
  \setlength{\columnseprule}{\indexrule}%
  \setlength{\columnsep}{\indexcolsep}%
  \chapter*{\indexname}
  \markright{\indexname}
  \preindexhook
  \setlength{\columnsep}{1.8em}%
  \begin{multicols}{3}}%
```



## 찾아보기

<b>E</b>	거시경제균형, 30	시장청산, 27
equilibrium business cycle theory, 27	경기변동이론 실물적, 28	o
	공급충격, 28	열린구간, 32
<b>F</b>	교란요인, 27	열린덮개, 35
Friedman, M., 28, 29	국소적 개념, 31	응골집합, 35
	규칙, 28	
<b>L</b>	균형경기변동이론, 27, 28	z
Lucas, R.E., 27, 29	극소곡면, 31	접벡터, 33
		정칙곡면, 32, 36

그림 5.1 이 책의 찾아보기 스타일

```
\linespread{1.2}\small
\small
\spaceskip=.275em plus .1em minus .3pt
\flushcolumns
\indexmark
\phantomsection
\addcontentsline{toc}{chapter}{\indexname}%
\parindent\z@
\parskip\z@ \@plus .3\p@ \relax
\let\item\@idxitem}%
{\end{multicols}
}

\renewcommand{\@idxitem}{\par\raggedright\hangindent 20\p@}
```

multicol 패키지를 이용하여 3단 조판하였다. pgreenbooksample 문서의 찾아 보기는 그림 5.1과 같은 모양이다.

ind 생성은 xindy를 이용하였다. 다음 명령으로 실행한다.

```
texindy -C utf-8 -L korean <main>.idx
```

# CHAPTER 6

## 정리류

### 6.1 정리, 명제, ...

수학식과는 달리 텍스트(와 수식)로 이루어져 있지만 수학적으로 특별한 의미를 지니는 문단을 ‘정리류 문단(theorem-like paragraphs)’이라 한다. 명제(proposition), 정리(theorem), 정의(definition), 보조정리(lemma), 따름정리(corollary), 주의(remark) 등이 여기에 속하며 증명(proof)도 특별한 정리류 문단이다.

일반적인 논문이라면 다음과 같은 amsthm이 제공하는 정리류 문단이 적절하겠으나,

**정리 1 (중심극한정리).** 서로 독립이며 동일한 분포를 따르는 확률변수  $X_1, X_2, \dots, X_n$ 에 대해, 각각의 평균은  $E(X_i) = \mu$ 이고 각각의 표준편차는  $\sigma$ 라 하자.  $\xi_n = \frac{\sum_{i=1}^n X_i - n\mu}{\sqrt{n}\sigma}$  라 둘 때,  $\xi_n$ 은 표준정규분포로 분포수렴한다.

단행본에서는 이렇게 해서는 시각적으로 잘 눈에 들어오지 않기 때문에(특히 한글 책에서) 박스를 치거나 문단에 장식을 하여 두드러지게 만드는 관행이 있다. 이 책에서 thm 환경을 제공한다.

**정리 6.1 (중심극한정리)** 서로 독립이며 동일한 분포를 따르는 확률변수  $X_1, X_2, \dots, X_n$ 에 대해, 각각의 평균은  $E(X_i) = \mu$ 이고 각각의 표준편차는  $\sigma$ 라 하자.  $\xi_n = \frac{\sum_{i=1}^n X_i - n\mu}{\sqrt{n}\sigma}$  라 둘 때,  $\xi_n$ 은 표준정규분포로 분포수렴한다.

이 환경을 corollary나 lemma로 확장할 수 있음을 보여주기 위해 cor와 lem도 (예제로서) 정의되어 있다. 이 둘은 모양은 같고 표제만이 다르다.

**따름정리 6.2** 서로 독립이며 동일한 분포를 따르는 확률변수  $X_1, X_2, \dots, X_n$ 에 대해, 각각의 평균은  $E(X_i) = \mu$ 이고 각각의 표준편차는  $\sigma$ 라 하자.  $\xi_n = \frac{\sum_{i=1}^n X_i - n\mu}{\sqrt{n}\sigma}$ 라 둘 때,  $\xi_n$ 은 표준정규분포로 분포수렴한다.

**보조정리 6.3** 서로 독립이며 동일한 분포를 따르는 확률변수  $X_1, X_2, \dots, X_n$ 에 대해, 각각의 평균은  $E(X_i) = \mu$ 이고 각각의 표준편차는  $\sigma$ 라 하자.  $\xi_n = \frac{\sum_{i=1}^n X_i - n\mu}{\sqrt{n}\sigma}$ 라 둘 때,  $\xi_n$ 은 표준정규분포로 분포수렴한다.

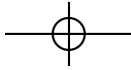
## 6.2 박스 디자인에 대한 코멘트

pgreenbook은 이들 박스를 아주 심플한 framed와 shaded만으로 구현하였다.

다른 책의 디자인을 보면 더 복잡하고 예쁜 정리류 박스 문단을 심심찮게 발견할 수 있는데, L<sup>A</sup>T<sub>E</sub>X에서는 tcolorbox 패키지나 tikz를 이용하여 얼마든지 원하는 박스 형식의 정리류 문단을 만들 수 있다. 심지어 tcolorbox 패키지의 부속 라이브러리 중에는 theorem이라는 것도 있다.

항상 그렇지만, 과도한 박스형 문단의 사용은 골치아픈 문제를 수반한다. 가장 대표적인 것이 “잘라지는 박스”의 문제이다. 박스가 잘라지지 않으면 페이지의 아랫쪽이 행하니 비어서 곤란한 경우가 생기지만 막상 이것을 분질러놓으면(tcolorbox로는 잘라지는 박스를 만드는 것이 어렵지 않다) 어떻게 해도 안 예쁜 결정적인 문제가 생긴다.

그리고 박스 안의 내용보다 박스 형태에 더 눈길이 가는 디자인은 곤란하다. 박스는 거의 신경쓰이지 않을 정도로, 단지 다른 문단과 구별이 되도록 하는 정도의 자기 주장이 강하지 않은 디자인이 좋은 디자인이다.



## 찾아보기

【ズ】	
정의.....	20

