



Oblivoir와 TikZ로 해보는 나만의 서식





- 제목, 저자, 날짜: `\title{·}`, `\author{·}`, (`\date{·}`)
 - `\thetitle`, `\theauthor`
 - 오늘 `\today` ⇨ 2020년 5월 30일
 - `\and`, `\thanks{·}`
- `\maketitle` ⇨ `pagestyle <title>`.
- `<titlingpage>` ⇨ `pagestyle <titleingpage>`
 - `\titlingpageend{·}{·}`
 - `<titlingpage*>`

(1) Oblivoir 클래스는 article 모드와 book 모드가 있다. 디폴트는 article이다. book 모드로 만들려면 `[chapter]` 옵션을 부여하여야 한다.

```
\documentclass[chapter]{oblivoir}
```

(2) article은 표제, 저자명, 날짜로 간단한 title을 만든다.

- 저자명 아래 소속(affiliation)을 나타내는 문단을 따로 두는 경우도 있다.
- 저자명에 표지(*, †, etc)를 붙이고 이것을 각주처럼 페이지 하단에 소속이나 고지사항을 적는 경우가 있다. 이것을 `thanksnote`라고 부른다. `\footnote`와 유사한 형태이다.
- `<abstract>`로 만들어지는 요약문까지를 타이틀로 보기도 한다. 단행본에서는 `abstract`가 존재하지 않는다.
- `\maketitle` 명령은 그 이전에 주어진 `\title`, `\author`, `\date` 정보를 이용한다. 이 정보들은 `\thetitle`와 `\theauthor`, `\thedata`에 저장되어 있다.¹ `\title` 등의 명령은 preamble에 두어도 되고 `\maketitle` 앞에 두어도 된다.
- `\maketitle` 명령이 주어지는 페이지는 페이지 스타일이 `<title>`이고 기본값은 `<plain>`과 같다. 즉 페이지 하단 중앙에 페이지 번호가 찍히는 것이다.
 - ▶ 타이틀 페이지에 페이지 번호가 찍히지 않게 하려면 어떻게 할 수 있겠는가?
- `\maketitle` 명령과 그 연관 명령은 클래스(주로 저널 기고를 위한)에 따라 다르며 청해진 대로의 양식으로 써야 한다. 이것을 사용자화하는 것은 쉽지도 않고 좋은 생각도 아니다.

(3) memoir와 같은 클래스로 단행본 규모의 ‘책’을 작성할 때나 보고서를 작성할 때는 이와는 다른 표제지가 필요하다. 단행본의 경우에는 book cover라고 부르는 외표지가 존재하는데 이것은 별도의 디자인으로 만드는 것이다. 다만 내표지는 L^AT_EX으로 작성할 수 있다.

- `<titlingpage>` 환경은 내표지를 (마음대로) 작성하기 위한 환경이다.

¹`\date` 명령은 생략할 수 있고 이 때는 오늘 날짜가 사용된다. 따라서 컴파일할 때마다 날짜가 바뀐다. 그리고 이 경우 `\thedata`는 정의되지 않는다.



- (내)표지는 주로 문서의 첫머리에 온다.
- $\langle titlingpage \rangle$ 환경이 끝날 때 페이지가 나누어지기 때문에 이 환경은 그 자체로 한 페이지를 구성하며, 원칙적으로 페이지 번호를 붙이지도 카운트하지도 않는다. 이 펫 페이지 표제는 홀수면(recto)이다. 표제가 끝나고 본문이 시작되면 거기서부터 1페이지이다.
- 양면문서(twoside)에서는 표제지 뒤의 짝수쪽(verso)을 비우기 때문에 백면 한 페이지가 생겨난다. 단면문서(oneside)에서는 그렇지 않다.
- $\langle titlingpage \rangle$ 도 본문의 일부로 쳐서 페이지를 매기게 할 수 있다. 그리고 싶다면 별표를 붙여서 $\langle titlingpage^* \rangle$ 환경으로 작성한다.²
- `\titlingpageend`라는 중요한 명령이 있다. 이것은 $\langle titlingpage \rangle$ 환경이 끝날 때 무슨 동작을 할 것인지 정해주는 것이다. 기본적으로
`\titlingpageend{\cleartorecto}{\clearpage}`
 와 같이 정의되어 있으며 이것이 바로 앞에서 설명한 대로 양면문서에서는 무조건 새 홀수쪽으로, 단면문서에서는 바로 다음페이지로 페이지 나누기를 하도록 한 것이다. 이 명령을 이용하여 동작을 조정할 수 있다.
 ▶ *titlingpage* 환경 다음에 마치 *article*처럼 페이지가 나누어지지 않고 바로 본문이 시작되게 하려면 어떻게 하면 되겠는지 답해보아라.
- $\langle titlingpage \rangle$ 는 $\langle titlingpage \rangle$ 라는 페이지 스타일을 해당 페이지에 적용한다. 기본적으로 아무 것도 찍지 않는 것($\langle empty \rangle$)이다.

⁽⁴⁾ $\langle titlingpage \rangle$ 안에서는 그 전체 페이지를 하나의 캔버스로 생각하여 마음대로 요소를 배치하면 된다. 이 때 TikZ를 잘 활용하면 좋다.

- 표제 페이지에서는 장식적 폰트를 임의적으로 (*ad hoc*) 사용해도 좋다. 어차피 표제지는 문서의 논리적 일부가 아니기 때문에 장식적으로 접근할 수 있다.
- 그림이나 소속 기관의 로고 등을 잘 배치하여 설득력있고 촌스럽지 않으며 심플하지만 아름다운 표지를 만들어보아라.
- 이 문서의 표제지에서는 `pgfornament`라는 것으로 귀퉁이에 장식을 넣었다. 폰트는 신영복체를 사용하였다. \langle 처음처럼 \rangle 이 생각나는 서체이다.

⁽⁵⁾ *article*의 `\maketitle`은 바꿀 수 없다³고 하였다. 그러나 $\langle titlingpage \rangle$ 환경을 이용하면 *article*에서도 자신만의 표제를 작성할 수 있다. 제목, 저자명, 날짜 순으로 배치하고 이 전체를 `framebox` 안에 넣은 *article* 모드의 `oblivoir` 표제를 작성해보아라. 표제가 끝난 위치에서 페이지가 나누어지지 않아야 한다. (이때, `\thanks` 주석의 처리는 조금 고민이 필요하다. 이것은 L^AT_EX을 더 잘 알게 되면 하기로 하고 우선은 무시한다.)

²이 문서가 그러고 있다. 표제 다음 쪽이 몇 페이지인지 확인하라.

³또는, 원래 쉽게 바꾸어 쓸 목적으로 만들어진 것이 아니다.



절 표제 번호붙이기

- `\book, \part, \chapter`
- `\section, \subsection, \subsubsection, \paragraph, \subparagraph`
- `\maxsecnumdepth{·}`, cf. `\setsecnumdepth{·}`
- `\counterwithin{·}, \counterwithout{·}`
- `\thesection`
- `\hangsecnum, \defaultsecnum`

(1) 장절명령은 `\book`에서 `\subparagraph`까지 정의되어 있다.

- Oblivoir에 `[chapter]` 옵션을 주면 `\chapter` 명령이 제대로 동작한다. `\part` 명령은 이 옵션과 상관없다.⁴ `\book`은 여러 책을 합쳐서 한 권으로 만들 때 쓰는 것으로 현실적으로 이 명령을 쓸 가능성은 극히 드물 것이다. 장(chapter) 이상의 document division은 추후 다루기로 하고 여기서는 `\section` 이하의 장절 편제에 대해서 살펴본다.
- `\subsubsubsection`은 없는 걸까? 없다. 사용자가 이 명령을 정의해서 쓰는 것은 가능하나, 그렇게 하지 말 것을 강력히 권고한다. 그래서 그 방법도 소개하지 않는다.

(2) section-류 명령은 왜 필요한가? 예를 들어 다음 두 코드가 만들어내는 결과를 비교해보자.

```
\noindent\textbf{\Large 1 행렬식}           \section{행렬식}
```

- 모양만 언뜻 보면 큰 차이 없어 보인다. 그러나 절의 표제를 왼쪽과 같은 방법으로 코딩해서는 절대로 안 된다.
- 오른쪽과 같이 장절명령을 사용한 문서구획은 *semantic* 코딩이다. 나중에 절 표제가 display 되는 방법을 얼마든지 바꿀 수 있다. lshort-ko에 보면, “그걸 다 찾아서 일일이 바꿀래?” 라고 하는 부분이 있다.
- 장절명령의 가장 중요한 의미는 TOC (table of contents)를 자동으로 만들기 위한 조치라는 점이다. 그뿐 아니라, 페이지의 상(하)단에 문서 절표제를 면주에 적어주기 위해서도 필요하다.

(3) Oblivoir로써 아무런 설정 없이, 문서를 작성해보면, `\section`과 `\subsection`까지 1.1과 같은 모양의 번호가 붙는다. 그 아래 수준에 대해서는 번호가 붙지 않는다. `[chapter]` 옵션이 있을 때와 없을 때 번호가 붙는 방식을 비교해보자.

- `subsection`의 번호가 1.1일 때, 점을 기준으로 그 앞의 것은 `section`의 번호이고 그 뒤가 `subsection`의 번호이다. `chapter`가 있을 때는 1.1.1과 같은 모양이 된다.
- 문서 내에서 어느 수준까지 절표제에 번호를 붙일 것인가 정해주려면 `\maxsecnumdepth`로써 선언한다. 인자로는 예컨대 `\maxsecnumdepth{paragraph}`와 같이 섹션 수준의 이름을 주면

⁴논리적으로 `\part`는 `\chapter`의 상위 장절명령이지만 실제로 이 둘은 분리되어 있어서 `\part`가 바뀌었다고 해서 `\chapter`가 다시 1장부터 시작하는 것이 아니다.



된다.

▶ `\maxsecnumdepth`를 `chapter`로 하면 무슨 일이 일어날까?

- (4) 장절 번호는 상위 레벨의 장절과 “연결” 되어 있다. 예를 들면 `section`이 시작되면 그 아래 수준의 장절 번호들이 모두 리셋된다. 이 “연결” 상태를 `\counterwithin`과 `\counterwithout` 명령으로 조절할 수 있다. 이름 자체가 의미를 다 보여주므로 추가 설명을 필요로 하지 않는다. `section`의 카운터를 `chapter`와는 따로 놀게 만들려면 다음과 같이 한다.

```
\counterwithout{section}{chapter}
```

- 이 명령에 별표를 붙이면 장절 번호의 모양은 그대로 두고 번호의 리셋 상태만 변경한다.
- 그렇다면, 다음 두 줄의 명령은 도대체 무슨 결과를 가져올 것인가?

```
\counterwithout{subsection}{section}
\counterwithin*{subsection}{section}
```

직접 실행하여 확인하여 보아라.

- (5) `\thesection`과 같은 종류의 매크로는 장절 번호를 가지고 있다. 나중에 페이지스타일 등을 작성할 적에 필요할지 모른다.
- (알 필요없는 정보) 옛날에, 절 번호의 끝에 마침표를 찍고 싶어하는 사람들이 존재했다. 이들을 위해서 `\thesection` 끝에 마침표를 붙이라는 답변이 붙기도 했다. 그러나 memoir, oblvioir 세계에서는 `\thesection` 류 명령을 사용자가 `\renewcommand`하는 것은 하면 안 된다.
 - 그러면 숫자 뒤에 마침표 붙이는 방법은? 그건 조금 후에 알아보자. (페이지 7을 보라.)
- (6) `\hangsecnum`은 절 표제를 식자할 때 번호 부분을 판면 바깥으로 끌어낸다. 그 반대(즉 default)는 `\defaultsecnum`이다. 아주 조금이지만 판면이 모던해보이는 효과가 있는지도. 직접 해보자.
- (7) 기본적으로 조판되는 결과를 보면, `\subsubsection`까지는 장절의 표제가 별도의 문단처럼 식자되는 데 비해(별행표제), `\paragraph` 이후 장절 명령은 글자가 조금 굵거나 들여쓰기 될 뿐으로 이어지는 문단 본문이 표제에 바로 이어서 같은 문단으로 식자된다. 후자를 파고들기 표제(*run-in title*)라고 한다. 원한다면 어떤 레벨의 장절 명령도 모두 파고들기 표제로 만들 수 있으나 그 방법은 다음 절에서 살펴본다.
- (8) KTUG 사설저장소에서 받을 수 있는 `kocircnum`이라는 패키지의 사용설명서에는 절 제목을 다음 처럼 하는 예가 소개되어 있다. TikZ를 활용하여 `section number`를 장식한 보기이다. (그러나 절 표제를 이런 식으로 넘버링하는 문서는 아마 찾기 힘들 것이다.)

8 기타



절 표제 간격

- `\setbeforesecskip`, `\setaftersecskip`
- `\setsecindent`

- (1) `beforesecskip`은 수직간격이다. 기본 `\baselineskip`에 이 값을 더한 것이 이전 문단과 새 절표제 사이의 간격이다.
- (2) `secindent`는 판면의 왼쪽 끝에서 절표제 (번호가 있다면 그 번호의 시작 위치)까지의 들여쓰기 값이다. 기본적으로 0pt이다.
- (3) `aftersecskip`의 값은 양수이면 수직간격이다. 이후의 따라붙는 문단과 별행으로 표제가 식자된다. 만약 이 값이 음수이면, run-in 표제로서 식자되며 그 절댓값에 해당하는 것이 표제와 문단 시작 사이의 수평 간격이다.
 - 이제 `\paragraph`와 `\subparagraph`가 파고들기 표제로 식자되는 이유가 `afterparaindent`가 음수이기 때문을 알았다. 이 값을 양수로 설정하면 별행표제가 되는 것이다. `subpara`도 마찬가지.
- (4) `beforeskip`도 양수이거나 음수일 수 있다. 이 값이 음수이면 별행 표제일 적에 그 뒤에 이어지는 첫 번째 문단을 들여쓰기 하지 않는다. 양수이면 그렇지 않다. 즉 `\section`, `\subsection`, `\subsubsection`은 모두 `beforeskip`이 음수이다.
 - 표제 다음 첫 행을 들여쓰기 하지 않는 것은 매우 보수적인 예전 관행이다. 그러나 아직도 (서양에서는) 학술 문헌의 경우 이 관행을 따르고 있다. 논리적으로 이미 표제가 있기 때문에 문단 시작 표지를 다시 두는 것은 잉여적(*redundant*)이다.
 - 첫 문단 포함하여 모든 문단의 들여쓰기를 허용하려면 (section 등의 `beforeskip`을 재설정해도 되지만) 간단히 `incentfirst` 패키지를 쓰면 된다. 이 패키지가 하는 일이 이 값을 양수로 바꾸어주는 것이다.
- (5) `skip`이라는 이름이 붙은 길이 변수는 `plus`와 `minus`로 그 허용치를 지정할 수 있다. 페이지의 아래 끝을 맞추려면 문단 사이 간격이 가변적이어야 한다. 예를 들면 다음과 같이 한다.


```
\setbeforesecskip{3.5ex plus 1ex minus .2ex}
```

 이리하여 실제 길이를 l 이라 하면 $3.3ex \leq l \leq 4.5ex$ 의 범위에서 가변적으로 그 길이가 변한다. 그러나 되도록 $3.5ex$ 에 가깝게 유지하려고 할 것이다.
- (6) 이 명령군은 `sectioning level name`을 약어로 줄여서 쓰고 있다. 각각 `sec`, `subsec`, `subsubsec`, `para`, `subpara`로 쓰는데 의미는 명확하다. 명령의 예시에서 밑줄이 그어져 있는 `sec`은 모두 이런 의미이다.
- (7) 장절 명령 간격의 기본값은 *memoir manual*, p. 95를 보라.



- `\setsecnumformat`
- `\setsecheadstyle`
- `\setsechook`

(1) 주의: `\setsubsecnumformat`과 같은 식의 명령은 없다. 이 `\setsecnumformat` 명령은 모든 section-류 명령의 number format을 설정한다. 사용법은 다음과 같다.

```
\setsecnumformat{\csname the#1\endcsname\quad}
```

- 따라서 모든 section-류 명령의 숫자 뒤에 마침표를 찍으려면
`\setsecnumformat{\csname the#1\endcsname.\quad}`
와 같이 한다.

- `\csname the#1\endcsname`이라는 것은 이를테면 `\thesection`, `\thesubsection`과 같은 각각을 모두 가리키는 것이라고 이해해두자.

(2) 그런데 만약 section에만 format을 적용하고 다른 장절표제에는 그렇게 하지 않으려 한다고 하자. 그럴 때는 어떻게 할 것인가?

- `\setsechook`은 장절 표제 식자를 시작하기 전에 어떤 원하는 코드를 삽입할 수 있게 한다. 이를 이용하면 된다. 이때 주의할 점은 #1이 아니라 ##1으로 해야 한다는 점과,⁵ 이 명령의 영향이 이후까지 미친다는 점이다. 그러므로 subsection의 경우에는 이 적용을 받지 않도록 해주는 것이 필요하다. 따라서,

```
\setsechook{\setsecnumformat{\csname the##1\endcsname.\quad}  
\setsubsechook{\setsecnumformat{\csname the##1\endcsname\quad}}
```

- ▶ *section*의 번호에만 `\fbox`로 프레임을 그려보아라.

(3) `\setsecheadstyle`은 주로 절 표제 문자열의 폰트를 지정하기 위한 것이다. 예를 들어 section의 타이틀을 `sffamily`로 적고 싶을 적에는

```
\setsecheadstyle{\sffamily\Large}
```

와 같이 할 수 있다.

⁵다른 명령의 인자 안으로 들어가서 정의되기 때문이다.



- 페이지 레이아웃
 - `\stockwidth`, `\stockheight`, `\paperwidth`, `\paperheight`
 - `\textwidth`, `\textheight`
 - `\linewidth`, `\hsize`
 - `\spinemargin`, `\foremargin`
 - `\setheadfoot`
- 용지와 마진
 - `\usepackage{fapapersize}`, `\usefapapersize`, `\usefastocksize`
 - `\oddsidemargin`, `\evensidemargin`
 - `\setmarginnotes`
 - `\marginparmargin`

(1) 표준 L^AT_EX 클래스와 마찬가지로 memoir도 기본 용지는 letterpaper이다. A4paper보다 너비는 조금 넓고 길이는 짧다.

- memoir는 매우 다양한 용지 옵션을 가지고 있다. 우리에게 필요한 것은 a4paper 정도이다.
- stock을 별도로 설정하는 경우라면, 반드시 [showtrims] 옵션을 주고 preamble에서 `\trimLmarks`를 적어두자. 재단선을 보여준다.⁶

(2) fapapersize 패키지를 사용하여 페이지 크기를 잡아보자. 이 문서는 `\usefapapersize{190mm,258mm,20mm,*,30mm,25mm}`

로 설정한 신국판 판형이다. 현재 상태에서 각 판면 길이들을 알아보면,

| | | | | | |
|-------------|--------------|-------------|--------------|----------------|--------------|
| stockwidth | 189.99777 mm | paperwidth | 189.99777 mm | textwidth | 149.72012 mm |
| stockheight | 257.99696 mm | paperheight | 257.99696 mm | textheight | 205.90129 mm |
| headheight | 4.21747 mm | headsep | 6.34991 mm | uppermargin | 29.99963 mm |
| footskip | 8.89005 mm | lowermargin | 22.09604 mm | marginparwidth | 15.07938 mm |
| spinemargin | 19.99976 mm | foremargin | 19.99976 mm | marginparsep | 2.46019 mm |

이 각 길이들이 어떤 의미인지를 *memoir manual*, pp.10–11에서 확인하라.

(3) article이나 report는 단면으로 작성해도 상관없다. 그러나 책은 반드시 양면(twoside)으로 만든다. 오른쪽을 홀수쪽(recto), 왼쪽을 짝수쪽(verso)이라 한다. 점잖은 책은 홀수쪽에서 새로운 chapter가 시작한다. 요즘은 짝수쪽에서 시작하는(방정맞은) 책이나 아무 데서나 시작하는(정신나간) 책들도 많지만.

⁶memoir의 페이지 metric과 pgf의 페이지 metric이 조금 달라서 stock을 활성화했을 때 TikZ에서 페이지 내 위치의 계산이 조금 어긋나는 경우가 있다. 이것은 주로 page node의 상대 위치 계산에서 발생하는 일이고 이에 대한 여러 해결책이 있으나, 위에 열거한 변수를 적절하게 계산하면 대부분의 경우 문제를 쉽게 해결할 수 있다.



- 그러나 article같은 작은 문서라 하더라도 왼쪽과 오른쪽의 쪽 모양(pagestyle)을 달리 설정하려면 —예컨대 페이지 번호가 짝수쪽에서는 왼쪽에, 홀수쪽에서는 오른쪽에 오거나, 짝수쪽 면주에는 책 이름, 홀수쪽 면주에는 절 이름을 넣는 따위— 문서를 [twoside]로 작성해야 한다.
 - 작은 규모의 문서에서는 foremargin과 spinemargin을 동일하게 작성해도 좋다. 보통 그렇게 한다.
- (4) 마진에 주석을 달거나 그림을 넣는 등 보충적 정보를 기록하는 방식의 조판이 유행하고 있다. 주로 교과서 같은 종류의 책에서 그리 하는데, 이를 위해서 margin 영역에 박스를 둘 수 있어야 한다.
- memoir는 \marginpar, \sidebar, \sidepar, \sidefoot이라는 네 종류의 마진 영역 활용 장치를 가지고 있다. 이 가운데 \marginpar만이 표준 L^AT_EX 클래스에 있는 것이다.⁷
 - \setmarginnotes{·}{·}{·} 명령은 \marginparsep, \marginparwidth, \marginparpush를 한꺼번에 설정하게 해준다. \marginparsep은 텍스트 블록과 marginpar 사이의 간격이고 \marginparwidth는 marginpar를 두는 박스의 너비이며, \marginparpush는 아래 위로 놓이는 marginpar 사이의 최소 간격이다. 모두 길잇값으로 지정한다.
 - margin과 marginpar는 서로 다른 것이다. margin은 텍스트 블록의 오른쪽과 왼쪽에 놓이는 여백 자체를 가리키는 말이고 marginpar는 이 margin 영역에 오는 텍스트 등의 stuff을 넣은 (플로팅) 박스를 가리키는 말이다. 따라서 예컨대 현재의 spinemargin이 19.99976 mm이라 하더라도 marginparwidth는 15.07938 mm일 수 있다. 보통 margin ≤ marginparwidth + marginparsep이 되도록 길이를 잡는다.
 - marginpar는 어디에 놓이는가? 양면 문서에서 홀수 페이지의 오른쪽, 짝수 페이지의 왼쪽이 기본값이다. 즉 spinemargin에 marginpar가 놓인다. 이 위치를 지정할 수 있는데, 이때 \marginparmargin{·}이라는 명령을 쓴다. 인자로는 left, right, outer, inner 넷 중의 하나. 기본값은 outer인 것이다. marginpar를 단면이건 양면이건 무조건 오른쪽 여백에만 두려 한다면 \marginparmargin{right}라고 하는 것으로 충분하다.
- (5) 어떤 책을 보면, 양면 조판인데도 불구하고 짝수쪽과 홀수쪽의 마진 영역이 같은 경우가 있다. 즉, 항상 텍스트 블록의 왼쪽편에 그림이나 주석이 놓이는 것이다. 이것을 구현하려면 다음 한 줄이 필요하다.
- ```
\setlength{\evensidemargin}{\oddsidemargin}
```
- (6) 연습문제: 『맨큐의 경제학』 제8판 한글 번역본의 펼침면 디자인을 실제 길이를 자로 재어서 구현하라. 페이지 사이즈, 텍스트 블록의 사이즈, 마진의 사이즈를 정확하게 설정하라. (할 수 있다면 section style도 비슷하게 해보면 좋겠지만.) 그외의 다른 요소는 무시한다. 텍스트는 아무 텍스트나 채워서 3페이지 이상을 만들어라.

<sup>7</sup>이 가이드에서는 \sidebar, \sidepar, \sidefoot에 대해서는 다루지 않는다. memoir manual의 해당 항목을 참조하라.



- 정렬
  - `\raggedright`, `\raggedleft`, `\centering`, `\par`
  - `<flushleft>`, `<flushright>`, `<center>`
- 리스트
  - `<enumerate>`, `<itemize>`, `<description>`
  - `\item` `\firmlist`, `\tightlist`, `\firmlists`, `\tightlists`
  - `\topsep`, `\labelwidth`, `\labelsep`, `\itemindent`, `\parsep`, `\listparindent`, `\leftmargin`, `\rightmargin`
- 리스트형 문단
  - `<verse>`, `<quote>`, `<quotation>`, `<adjustwidth>`
- 매달린 문단: `\hangfrom`, `\hangpara`, `<hangparas>`

(1) 문단의 정렬에 대해서는 따로 설명할 것이 없다. *lshort-ko*에서 다 공부한 것이다. TeX은 justification (양끝맞춤)에 대한 집착이 매우 큰 조판 도구임을 잊지 말자. 특별한 목적이 아니라면 양끝맞춤을 정당한 것으로 받아들이도록 한다.

(2) 다만, “정렬 선언”과 “정렬 환경”의 차이를 이해해두는 것이 필요하다. 기본적으로 정렬 환경은 아래위에 약간의 수직 간격을 추가한다. 그리고 정렬 선언은 그 이후로 계속 영향을 미치기 때문에 범위를 설정({와 })하는 것이 중요하며, 정렬 선언이 효력을 발휘하는 것은 적어도 하나 이상의 문단이 flush되었을 때라는 것을 기억해야 한다. 그러므로 마지막 문단의 정렬을 위해 `\par`가 있어야 한다. 다음 예를 보자.

```
{\raggedright ... }
```

절망이 허망한 것은 희망이 그러한 것과 같다. 만약 내가 오히려 이 밝지도 않고 어둡지도 않은 희망 속에서 구차하게 삶을 부여잡고 견뎌내야 한다면 나는 반드시 저 사라져버린 서럽고 쓸쓸하며 어렴풋해 아득하기만 한 청춘을 찾아서 구해야 할 것이다.

```
{\raggedright ... \par }
```

절망이 허망한 것은 희망이 그러한 것과 같다. 만약 내가 오히려 이 밝지도 않고 어둡지도 않은 희망 속에서 구차하게 삶을 부여잡고 견뎌내야 한다면 나는 반드시 저 사라져버린 서럽고 쓸쓸하며 어렴풋해 아득하기만 한 청춘을 찾아서 구해야 할 것이다.

(3) 여기서 더 자세히 다룰 수는 없지만 `ragged2e`라는 패키지가 제공하는 몇 가지 환경을 공부할 필요가 있음을 지적해두겠다. `\raggedright`는 단어 단위로 행자름이 일어나지만 이 패키지는 그것을 하이프네이션이 가능한 곳에서 할 수 있게 하는 `\RaggedRight` 같은 종류의 명령과, 강제로 양끝맞춤하는 `\justify` 명령을 제공한다.

(4) 리스트 환경은 L<sup>A</sup>T<sub>E</sub>X의 문단 구조에서 가장 중요한 종류의 것이다. 잘 아는 `<enumerate>` 등은 모두 `<list>` 환경을 적절하게 재구성한 것이다. 이 환경의 모양을 사용자화하려면 *memoir manual*, p. 141



에 나오는 그림을 잘 이해하고 있어야 한다. 이 그림에 등장하는 여러 파라미터를 재정의하거나 설정함으로써 리스트 환경의 모양을 조절할 수 있다.

- (5) `<verse>`, `<quote>` 등은 모두 `<list>` 환경으로 만들어진 문단 구조이다. 혹시 이런 종류의 문단을 다루다가 `\item`이 없다는 에러를 만나는 수가 있는데 그럴 때는 문단 맨 첫머리에 `\item`을 두면 된다. 실제로는 이 환경의 정의 안에 한 개의 `\item`이 들어 있는 경우가 많다.
- (6) 참고로 memoir는 시집 조판을 가능하게 할 정도로 `<verse>` 환경이 크게 확장되어 있다. 주로 영문학상의 유명한 시를 조판하는 방식이 많이 반영되어 있다. 우리말 시는 이런 조판 기법과 약간 다르기 때문에 그다지 쓸 곳이 없지만.
- (7) 그림 1은 `<verse>` 환경을 사용하는 예를 보인다. `\poemtitle` 관련 명령으로는 시들만 모아서 목록을 만들 수 있기도 되어 있으나 더 논의하지 않겠다.
- (8) `<adjustwidth>`는 의외로 쓸모가 많은 환경이다. 그러나 대체로 평문단 사이에서는 훌륭하게 동작하지만 복잡하게 nested된 구조에서는 뜻대로 되지 않는 경우도 있다. 그것은 리스트 환경 자체의 문제이므로 각 상황에 맞게 대응하는 수밖에 없다. 문단의 좌우 폭을 임의의 길이만큼 들여쓰게 할 수 있다. 예를 들면,

```
\begin{adjustwidth}{1cm}{2cm}
```

절망이 허망한 것은 희망이 그러한 것과 같다. 만약 내가 오히려 이 밝지도 않고 어둡지도 않은 희망 속에서 구차하게 삶을 부여잡고 견뎌내야 한다면 나는 반드시 저 사라져버린 서럽고 쓸쓸하며 어렴풋해 아득하기만 한 청춘을 찾아서 구해야 할 것이다.

이와 같은 문단을 식자할 수 있고 좌우 들여(내어)쓰기 폭을 조절할 수 있다.

- (9) “매달린 문단”이란 첫 줄이 내어쓰기 되는 문단 모양을 가리킨다. 즉 문단의 `\leftmargin`이 첫 줄에 적용되지 않는 것을 말한다. `\hangindent` 매크로를 사용하는 전통적인 기법이 있고, memoir에는 `\hangpara`라는 명령이 있다. 후자가 좋은 점은  $n$ 줄 이상을 내어쓰기할 수 있다는 점이다. 그림 2. 이와 같은 문단 모양은 이미 문단 모양을 통제하고 있는 리스트 환경(즉 `adjustwidth`, `quote` 등) 내에서는 동작하지 않는다는 점을 알아두자. 리스트 환경 내에서는 문단 모양 관련 파라미터를 직접 바꾸어야 한다.
- (10) `\hangpara`의 재미있는 응용으로서, 한 문단의 처음 몇 줄을 들여쓰기할 수 있다. `hang`할 라인 수를 음수 값으로 주는 트릭이다. 그림 2의 세 번째 예를 보라.
- (11) `<hangparas>`는 여러 개의 문단에 이 작용을 적용할 수 있게 한 환경이다.
- (12) 좀더 즉흥적으로 쓸 수 있는 `\hangfrom`도 있다. 이 명령이 문단 첫머리에 오면 인자로 주어진 만큼을 내어쓰기하는 hanging paragraph로 식자한다. 역시 리스트 환경 내에서는 되지 않는다.
- (13) `<enumerate>` 환경은 옵션 인자로 item label의 모양을 바꿀 수 있게 되어 있다. 여기에 올 수 있는 것은 A, a, i, I, 1, 가, (가), ㉠, 7, (7), ⑦, (1), ①, (a), ㉢이다.



## 서시

죽는 날까지 하늘을 우러러  
한 점 부끄럼 없기를  
앞새에 이는 바람에도  
나는 괴로워했다.

별을 노래하는 마음으로  
모든 죽어가는 것을 사랑해야지  
그리고 나한테 주어진 길을  
걸어가야겠다.

오늘 밤에도 별이 바람에 스치운다.

```
\begin{verse}
\poemtitle{서시}
죽는 날까지 하늘을 우러러\\
한 점 부끄럼 없기를\\
앞새에 이는 바람에도\\
나는 괴로워했다.
```

```
별을 노래하는 마음으로\\
모든 죽어가는 것을 사랑해야지\\
그리고 나한테 주어진 길을\\
걸어가야겠다.
```

```
오늘 밤에도 별이 바람에 스치운다.
```

```
\end{verse}
```

〈그림 1〉 verse 환경

```
\hangindent=2em
```

절망이 허망한 것은 희망이 그러한 것과 같다. 만약 내가 오히려 이 밝지도 않고 어둡지도 않은 희망 속에서 구차하게 삶을 부여잡고 견뎌내야 한다면 나는 반드시 저 사라져버린 서럽고 쓸쓸하며 어렵풋해 아득하기만 한 청춘을 찾아서 구해야 할 것이다.

```
\hangpara{2em}{2}
```

절망이 허망한 것은 희망이 그러한 것과 같다. 만약 내가 오히려 이 밝지도 않고 어둡지도 않은 희망 속에서 구차하게 삶을 부여잡고 견뎌내야 한다면 나는 반드시 저 사라져버린 서럽고 쓸쓸하며 어렵풋해 아득하기만 한 청춘을 찾아서 구해야 할 것이다.

```
\hangpara{3em}{-2}
```

절망이 허망한 것은 희망이 그러한 것과 같다. 만약 내가 오히려 이 밝지도 않고 어둡지도 않은 희망 속에서 구차하게 삶을 부여잡고 견뎌내야 한다면 나는 반드시 저 사라져버린 서럽고 쓸쓸하며 어렵풋해 아득하기만 한 청춘을 찾아서 구해야 할 것이다.

〈그림 2〉 \hangindent와 \hangpara



- framed
  - `<framed>`, `<shaded>`, `<snugshade>`, `shadecolor`
  - `<qframe>`, `<qshade>`, `<leftbar>`
- tcolorbox
- theorem류 문단
  - `amsthm`
  - `tcolorbox`: `tcbtheorem`
- `<tabbing>`과 `\tab`

- (1) `<framed>`류 환경은 framed 패키지의 것과 거의 같다. 다만 framed에는 있는 `<oframed>`가 없을 뿐이다.
- 이 뭔가 없어보이는 프레임 박스는 한때 페이지 사이가 찢어진다는 점에서 혁신적인(!) 것으로 대접받은 적이 있다. 지금도 그럭저럭 써먹을 만은 한데, 공들인 느낌이 전혀 나지 않으니까 눈에 띄는 문서를 만들 때는 쓰지 않아도 좋다.
  - `<shade>`류 환경은 허접한 색으로 바탕을 칠한다. 색상을 정해주려면 `xcolor` 패키지를 로드하고 `\colorlet{shadecolor}{yellow!40!red}`와 같이 `shadecolor`라는 이름의 색상을 지정해주면 된다.
  - 유사한 환경이나 그 응용에 대해서는 *memoir manual*에 상당히 자세한 설명이 있으니 꼭 필요하면 참고.
- (2) 박스나 색상이 칠해진 독립된 문단을 만들기 위한 패키지로 `mdframed`와 `tcolorbox`가 있다. 이 패키지들의 사용법을 소개하는 것은 이 글의 범위를 넘는다. 오늘날 많은 사람들이 `tcolorbox`를 추천하고 있다는 것을 적어둔다. TikZ의 강력한 응용 사례 중의 하나이다.
- (3) 수학적 문단, 즉 정리, 정의, 보조정리, 따름정리 등을 별개의 문단으로 식자하는 것이 수학적 문헌의 일반적 관습이다. 그 모양을 특별하게 꾸미는 여러 가지(적어도 10여가지) 패키지가 있으나, 클래식한 `amsthm`에만 의존하든가, 그렇지 않으면 `tcolorbox`의 `theorems` 라이브러리를 이용하는 것을 추천한다. 원하는 거의 모든 문단 모양을 만들 수 있을 것이다.
- (4) 전통적인 L<sup>A</sup>T<sub>E</sub>X의 `<tabbing>` 환경은 더이상 사용하지 않는 것이 좋다. 그 대신 `tabto` 패키지가 제공하는 `\tab`과 `\TabPositions`를 활용하도록 하자. 이를 이용하면 쓸데없이 `<tabular>`로 `tabstop`을 맞추던 예전의 관행도 줄일 수 있다.<sup>8</sup>

<sup>8</sup>`<tabular>`는 매우 중요한 도구이지만 이런 목적으로 사용하기에는 사용상의 실수를 부를 가능성이 높다.



- predefined pagestyles
  - `⟨empty⟩`, `⟨plain⟩`, `⟨headings⟩`, `⟨myheadings⟩`, `⟨simple⟩`, `⟨ruled⟩`, `⟨Ruled⟩`, `⟨companion⟩`, `⟨book⟩`, `⟨chapter⟩`, `⟨cleared⟩`, `⟨part⟩`, `⟨title⟩`, `⟨titlingpage⟩`; `⟨hanguk⟩`
- `\coppypagestyle{·}{·}`, `\aliaspagestyle{·}{·}`
- `\makepagestyle{·}`
- header and footer
  - `\makeoddhead{·}{·}{·}{·}`, `\makeevenhead{·}{·}{·}{·}`
  - `\makeoddfoot{·}{·}{·}{·}`, `\makeevenfoot{·}{·}{·}{·}`
- `\thepage`, `\rightmark`, `\leftmark`, `\normalrulethickness`
- `\makeheadrule{·}{·}{·}`, `\makefootrule{·}{·}{·}`,
- `\makeheadfootruleprefix{·}{·}{·}`
- `\makerunningwidth{·}[·]{·}`
- `\makeheadposition{·}{·}{·}{·}{·}`
- `\makeheadfootstrut{·}{·}{·}`
- psmarks
  - `\makepsmarks{·}{·}`
  - `\createmark{·}{·}{·}{·}{·}`
- `\pagestyle`, `\thispagestyle`

(1) 업계 용어인 “하시라”는 기둥(柱)이라는 뜻이다. 표준 교과서 용어인 “면주(面柱)”<sup>9</sup>는 이 일본어를 있는 그대로 옮긴 것이다. 이 ‘주’가 “주석(note)”이라는 의미가 아님에 유의하라.

- 마이크로소프트 워드 한국어판에서는 이것을 “머리글/바닥글”이라고 한다. 한글에서는 뭐라는지 모르겠네.
- 그러나 ‘preface’나 ‘foreword’의 번역어도 “머리글”이라면 어찌라는 말일까?
- 엄밀히 말하면 면주와 페이지스타일이 동일한 것은 아니므로 혼용하기로 한다.

(2) 기정의 페이지스타일은 반드시 어떤 모양인지 확인하기 바란다. 이를 위해서는 적어도 적절한 분량의 텍스트와 더불어 장/절 편제가 포함되어 있어야 한다. 면주에 장(절)의 제목이 나타나기 때문이다.

<sup>9</sup>페이지를 뜻하는 ‘면’의 한자는 원래 頁이다. 그러나 이 글자를 ‘면’으로 읽던 옛 독법을 지금은 거의 잊어버렸다. 유력한 인터넷 한자사전은 이 글자의 뜻이 “페이지”인 것은 알지만 음이 ‘면’이었던 것은 알지 못해서 “페이지 頁”이라고 하고 있더라. 그런 사정으로 페이지를 가리키는 ‘면’에 해당하는 글자로 面을 (대신) 쓰고 있다.



《*hangu*》은 `oblivoir`에 의한 페이지스타일이고, 밑줄 그은 네 개는 표준  $\text{\LaTeX}$  클래스에도 있는 것들이다.

- (3) `\copypagestyle`과 `\aliaspagestyle`의 차이는 앞의 것이 수정 가능한 데 비해 뒤의 것은 불가능하다는 것이다. `\aliaspagestyle`이 쓰이는 대표적인 경우를 생각해보자. `\chapter` 명령이 불리면 그 페이지의 페이지스타일은 `chapter`가 된다. 그런데 `chapter` 페이지스타일은 `plain`과 동일하다. `\chapter` 명령이 불린 페이지 하단의 폴리오까지 지워서 아무 것도 표시되지 않게 하려면,

```
\aliaspagestyle{chapter}{empty}
```

이렇게 한다.

- (4) `\makepagestyle{}`는 인자로 주어지는 이름의 페이지 스타일을 새로이 만든다. 이름은 띄어쓰기 없고 아스키 문자로 된 평이한 이름을 지정하면 된다. 이 뒤로 해당하는 페이지스타일의 설정을 행한다.

- `odd/even`, `head/foot`, `left/center/right`. 단면문서라면 `odd`에 해당하는 `head`와 `foot` 설정만 행하면 된다. 그러나 양면문서라면 `odd`와 `even`에 각각 `head`와 `foot` 설정을 해야 한다. 그러면 네 개의 명령을 실행하게 되는데, 각 명령은 첫째 인자가 `pagestylename`이고 그 뒤의 세 개 인자가 각각 `left`, `center`, `right` 위치에 올 것을 정하는 것이다. `odd head`에 3개, `even head`에 3개, `odd foot`에 3개, `even foot`에 3개이므로 면주 설정을 할 수 있는 공간은 모두 12군데이다.
- 이를테면 `XX`가 `plain`과 비슷하게 페이지 하단에 페이지 번호만 찍지만 짝수쪽에는 왼쪽 끝에, 홀수쪽에는 오른쪽 끝에 둔다고 생각해보자.

```
\makepagestyle{XX}
\makeoddhead{XX}{-}{-}
\makeevenhead{XX}{-}{-}
\makeoddfooter{XX}{-}{\thepage}
\makeevenfoot{XX}{-}{\thepage}
\pagestyle{XX}
```

- 만약 헤더에 보노보노 그림을 넣고 선을 긋고 싶다면 ;-) 단순히 `\includegraphics`만으로는 위치를 조절하기가 어려울 것이다. `TikZ`가 좋은 해결책이다. `overlay`같은 옵션을 이용해서 머리글 부분을 하나의 캔버스라고 보고 그림을 그리듯이 원하는 요소를 배치하면 좋다.
- `\thepage`는 현재 페이지 번호이다.

- (5) `\leftmark`와 `\rightmark`는 특정한 문자열을 간직하고 있는 매크로이며 주로 장의 제목 또는 절의 제목이 들어 있다. `memoir`와 표준  $\text{\LaTeX}$  클래스는 `book`에 있어서 짝수쪽의 면주에는 `chapter`의 타이틀을 넣고, 홀수쪽의 면주에는 `section`의 타이틀을 넣는 것을 관행으로 하고 있다. 우리나라 북 디자인은 이와 조금 달라서 짝수쪽 면주에는 주로 책의 제목, 홀수쪽 면주에는 장(`chapter`)의 제목을 넣는데, 가만히 생각해 보면 이 면주의 정보는 자신이 지금 읽고 있는 곳이 어디인가를 가리키기 위한 것인데 거기에 책 제목이 들어 있어 본들 그게 무슨 추가적인 정보가 되겠는가? 썩 훌륭한 관행이라고 하기는 어렵다. 라이트노벨이라면 어차피 문서 구획이 장(또는 절) 한 수준밖에 없으니 그렇게도 할 수 있겠구나 싶지만. 요네자와 호노부의 『빙과』 한국어판(엘릭시르)도 그런 식으로 면주를 붙이고



있었다(재미있게 읽은 라이트노벨이어서 언급한다). 그러나 학술 서적에서는 아무래도 이상한 느낌이 있다.

- `\rightmark`와 `\leftmark`를 정의하는 것은 `\markboth{.}{.}`와 `\markright{.}`라는 명령이다. 그런데 이 명령은 `\chapter` 또는 `\section` 장절명령이 실행될 때 호출된다. 다르게 말하면, `\chapter` 또는 `\section`이 실행될 때에 `\rightmark`와 `\leftmark`에 일정한 문자열이 저장된다.
- 그런데, `\chapter` 또는 `\section` 명령이 불릴 때 `\markboth` 나 `\markright`를 어떤 식으로 호출할지를 정해주는 것은 `\chaptermark`와 `\sectionmark`라는 명령이다. 이를테면, `\sectionmark`는 예컨대 다음과 같이 되어 있다. (실제로는 더 복잡하지만 가상코드라고 이해하기 바란다.)

```
\newcommand*\sectionmark[1]{\markright{#1}}
```

그러므로 우리는 이 `\chaptermark`이나 `\sectionmark`를 재정의하여 `\chapter`나 `\section` 명령이 불릴 때의 동작을 제어할 수 있다.

- 더 나아가서, `\chaptermark`과 `\sectionmark`를 정의해주는 명령이 따로 있다. 그것이 `\createmark`라는 명령이다. 이 명령은 다시, `\makepsmarks`에 의하여 특정한 `pagestyle`이 설정되는 순간 (즉, `\pagestyle{.}` 명령이 호출되는 순간) 일괄적으로 `\chaptermark`와 `\sectionmark` 등을 새로이 정의한다.
- 따라서, 사용자는 `pagestyle`을 새로 정의하면서 이 페이지 스타일의 면주에 어떤 텍스트가 올 것인지를 `\makepsmarks`의 인자 안에서 `\createmark` 명령으로 설정해주면 되는 것이다. 이것으로 위에 언급한 모든 과정이 자동으로 처리된다.
- 원하는 면주 텍스트가 올라오지 않을 때의 극단적 처방으로 `\markboth`를 직접 실행하는 방법이 있지만 뭔가 개운치 않을 것이다.

<sup>(6)</sup> `\createmark{.}{.}{.}{.}`의 사용법을 알아두자. 이 매크로는 일단 `\makepsmarks{.}{.}`의 두 번째 인자 범위 안에서 불러야 한다. (첫 인자는 `pagestyle`의 이름이다.)

- 첫 번째 인자는 장절 명령의 레벨 이름이다. 예컨대 `chapter` 또는 `section`일 수 있으며 이 각각의 경우 `\chaptermark`와 `\sectionmark`를 재정의하겠다는 의미가 된다.
- 두 번째 인자에는 `left`, `right`, `both` 셋 중 하나가 올 수 있는데, 각각 `\leftmark`, `\rightmark`, 그리고 양쪽 다를 재정의한다는 의미이다.
- 세 번째 인자에는 `shownumber`, `nonumber`, `notitle` 셋 중의 하나가 온다. 말 그대로 `\leftmark` 나 `\rightmark` 매크로에 장절 번호를 포함시킬 것인가 여부이다. 세 번째의 `notitle`은 장절 번호만 넣고 표제 문자열을 넣지 말라는 (특별한) 의미이다.
- 네 번째와 다섯 번째 인자는 장절 번호 앞뒤에 무엇을 넣을 것인가 정하는 것이다. 만약 세 번째 인자의 선택이 `nonumber`라면 이 두 인자는 비워두면 된다.

<sup>(7)</sup> `\makerunningwidth{.}{.}`는 머리글/바닥글의 너비(`width`)를 제어한다. 첫 인자는 `pagestyle`의 이름이고 그 다음의 옵션 인자는 `foot`, 필수 인자는 `head`의 너비를 설정한다. 이 매크로는 주로



머리글/바닥글에 `\textwidth`보다 긴 너비를 갖는 선을 그리거나 그림을 넣거나 할 목적으로 사용한다. 기본적으로 `\textwidth`와 동일하게 설정되어 있다. 《*companion*》 스타일과 같이 `\textwidth`를 넘어서 마진까지 차지하는 긴 줄을 그을 때 필요하다.

- (8) `\makeheadposition{·}{·}{·}{·}{·}` 매크로는 면주 각 요소(12개)의 정렬 방식을 설정한다. 첫 인자는 `pagestyle`의 이름이고 그 뒤부터 차례로 `even head`, `odd head`, `even foot`, `odd foot` 순서이며 여기에 올 수 있는 인자는 `flushleft`, `center`, `flushright` 셋 중 하나이다. 기본적으로 `left=flushleft`, `center=center`, `right=flushright`로 설정되어 있다.

`header`의 `running width`가 `\textwidth`보다 긴 경우, 특히 `\pagewidth` 전체일 경우를 가정하자. 예컨대 홀수쪽 상단, 왼쪽 영역의 `position`이 `flushleft`라면 홀수쪽 상단 왼쪽 영역의 시작점이 텍스트 블록의 시작점과 일치한다는 의미이다. 면주의 텍스트가 텍스트 블록보다 왼쪽까지 나오도록 조판하고자 한다면 이 정렬 방식을 `center`나 `flushright`로 맞추어야 한다. 흔히 착각하기 쉬우므로 주의를 요한다.

- (9) `\rule` 관련된 명령은 설명을 줄이겠다. 왜냐하면 우리는 만약 면주 영역에 줄을 그을 일이 있으면 TikZ로 할 생각이니까. 아무튼 `memoir` 자체에 헤더/푸터에 줄을 긋고 그 줄의 두께를 조절하고 색칠을 하는 등 조작을 가하는 매크로 준비되어 있다는 인상 정도만 가져도 상관없다.

- (10) 마지막으로, 이렇게 고생하여 정의한 `pagestyle`을 문서에서 적용하는 일이다. 그것은 `\pagestyle{·}` 매크로를 호출하는 것이다. 이 매크로가 호출된 그 페이지부터 페이지 스타일이 지정된다.

`\thispagestyle{·}`은 딱 한 페이지만 특정 페이지 스타일을 적용하려 할 때 쓴다. 예컨대 `<titlingpage>` 환경은 그 페이지에 《*titlingpage*》 스타일을 `\thispagestyle` 명령으로 적용한다.

같은 페이지 안에서 이 매크로가 여러 번 불렸을 때 유효한 것은 동일 페이지 내의 마지막으로 불린 명령이다.



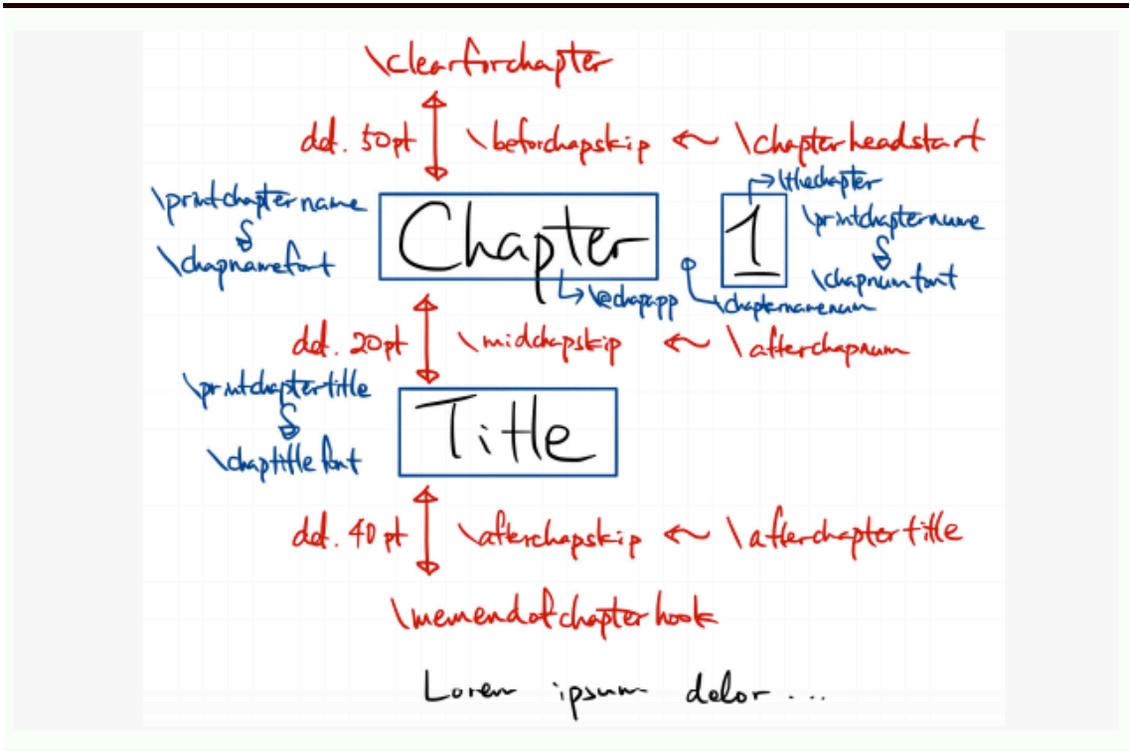
- floats
  - `<figure>`, `<table>`, `<marginpar>`
  - `\newfloat`
  - `\suppressfloats`, `\FloatBlock`
- caption
  - `\caption`, `\legend`
  - `\newfixedcaption`, `\captionof` of `CAPT-Of`
  - `\figurename`, `\tablename`, `\thefigure`, `\thetable`
  - `\captiondelim`, `\captionnamefont`, `\captiontitlefont`
  - `\captionstyle[·]{·}`
  - `\hangcaption`, `\normalcaption`, `\indentcaption{·}`
  - `\precaption`, `\postcaption`
- `[figtabcapt]`

- (1) *lshort-ko*에 “떠다니는 개체”라고 번역된 floating objects 또는 floats는 일종의 조판상의 장치이다. 텍스트의 흐름을 방해하지 않으면서 판면 한 쪽에 놓이는 objects들을 총칭하는 말이다.
- (2) 우리가 그림이나 표를 문서에 넣을 때, 이 그림이나 표가 놓이게 될 위치를 미리 예단할 수 없다. 좋지 않은 것은 이런 것들은 중간을 잘라서 다른 페이지로 쪼개지 못한다는 데 있다. 그러므로, “다음 그림과 같이”라든가 “아래 표를 보라”든가 하는 방식으로 문서를 작성하면 안 된다. 그 그림이 “아래”나 “다음”에 있지 않을 수 있고, 혹시 역지로 거기 갖다주면 판면이 엉망이 될 것이 틀림없다.
- (3) 표와 그림에 대한 것은 *lshort-ko*에서 중요한 것을 다 배웠을 것이다. memoir에 대해 말하자면, 표와 그림 말고 임의의 어떤 float를 정의하고, 이것을 떠다니게 하고, 여기에 caption과 label을 붙여 참조하고, 이것들의 목록(list)을 자동으로 작성할 수 있게 해준다. 흥미로운 기능이지만 상술하지 못하고 *memoir manual*을 공부할 것을 권할 수밖에 없다.
  - ▶ 『맨큐의 경제학』에서 어떤 것이 플로트로 처리되어야 할지 예상해볼 수 있겠는가?
- (4) 캡션은 그림이나 표에 붙이는 타이틀을 가리킨다. `\caption` 명령이 하는 일은 두 가지인데, 하나는 그림과 표에 번호를 활성화하여 `\label` 명령으로 그 번호를 저장할 수 있게 하고, 다른 하나는 캡션 텍스트(타이틀)을 모아서 목록(`\listoffigures`, `\listoftables`)을 만들 준비를 한다는 것이다. 그러므로 `\caption`을 붙이지 않은 그림은 `\label`을 달 수 없으므로 `\ref`로 참조할 수 없다. 그리고 원칙적으로 `\caption` 명령은 `<figure>`나 `<table>`과 같은 float 환경 안에 있어야 한다.
- (5) memoir에는 `\legend`라는 유사 `\caption` 명령이 있다 (범례라는 뜻이다).
- (6) 가끔 플로팅시키지 않은 그림에도 캡션을 달고 참조하고 싶을 때가 있다. 정상적이지는 않지만 이를 위한 장치는 준비되어 있다. memoir 자체에 `\newfixedcaption`으로 비-플로트 캡션을 새로



만들어 낼 수 있게 되어 있고, `capt-of` 패키지의 `\captionof` 명령에 의하여 플로팅 환경 안이 아닌 개체에도 캡션과 라벨을 달 수 있다.

- (7) `\figurename`과 `\tablename`은 각각 “그림”과 “표”라는 문자열이다. `oblivoir`에서 그렇게 정해두었다. 재정의할 수 있다. `memoir`나 표준  $\LaTeX$  클래스에서는 각각 “figure”와 “table”이다.
- (8) `\figure`와 `\table`의 번호는 `\thefigure`, `\thetable` 안에 들어 있다. `\caption` 명령이 실행되면 이 번호가 확정된다. 보통 장 번호와 연계되어 있어서 1.1같은 형식인데, 이미 배운 `\counterwithin`, `\counterwithout`을 이용하여 장번호가 아니라 다른 카운터와 연계하거나 아예 독립적인 카운터가 되게 만들 수 있다. `\section` 카운터를 다룰 때와 똑같다.
- (9) 캡션의 스타일링을 위한 매크로들이 많다.
  - `\captiondelim`은 “그림 1.1: 그림제목”에서 :에 해당하는 문자이다.
  - `\captionnamefont`, `\captiontitlefont` 이름 그대로이다.
  - `\captionstyle` 옵션 인자는 `short`, 필수 인자는 `long`의 경우에 캡션 문단을 식자하는 스타일을 지정한다. 예를 들어 `\captionstyle[\centering][raggedright]`은 캡션 타이틀이 짧으면 `centering`, 길면 `raggedright`으로 식자하라는 것이다.
  - `\hangcaption`은 첫 줄을 내어쓰기 한다. `\indentcaption`은 `indent` 값을 인자로 준다. 원래로 되돌리려면 `\normalcaption`.
  - `\precaption`과 `\postcaption` 매크로는 캡션 문단 전체의 시작 또는 종료 후의 처리에 필요한 것을 둘 수 있다.
- (10) 이밖에 캡션에 관한 많은 문제와 해결책이 있다. 특히 소논문 등에 자주 나오는 `subfigure`와 `subcaption`을 처리하는 방법도 *memoir manual*에 제시되어 있다. 여기서는 더 다루지 못한다.
- (11) `oblivoir`의 문서 옵션으로 `[figtabcapt]`를 주면, 캡션의 모양이 “<그림 1> 캡션 제목”과 같은 형태로 변한다.



- (1) chapter style은 memoir의 핵심적인 기능 가운데 하나이다. 공주대학교 워크숍에서 수차에 걸쳐 다루어졌다.
- 위의 그림은 2019년 워크숍에서 이재호 군이 발표한 “memoir chapter style”이라는 발표자료의 한 페이지이다. chapterstyle-beamer 자료를 참고하라.
  - 미리 정의된 chapter style이 여럿 있다. 이에 대해서는 위의 슬라이드와 memoir manual을 참고할 수 있다.
- (2) oblivoir는 이와 거의 동일하지만 조금 다른 방법으로 장 표제를 만든다. 이주호 선생의 <텍 원고 편집>이라는 공주대학교 문서작성 워크숍 2013의 발표자료<sup>10</sup>는 oblivoir의 장 스타일을 어떻게 꾸미는지 시각적으로 친절하게 보여주고 있다. (우리가 공부한 내용이 더 상세히 설명되어 있는 것도 있으므로, 이 슬라이드 자료를 반드시 읽어보기를 추천한다.) 요약하면,
- \printchaptername 매크로가 없다.
  - \prechapternum과 \postchapternum이 있다. 이것은 “제 1 장”의 ‘제’와 ‘장’에 각각 해당하는 매크로이다.
  - \hchaptertitlehead가 별도로 정의된다. 이것은 running heading에서 “제”와 “장”을 제대로 찍기 위한 것이다.

<sup>10</sup>[http://wiki.ktug.org/wiki/wiki.php/LaTeXWorkshop/2013?action=download&value=talk2\\_TeXnica1\\_Book\\_Design\\_by\\_progress.pdf](http://wiki.ktug.org/wiki/wiki.php/LaTeXWorkshop/2013?action=download&value=talk2_TeXnica1_Book_Design_by_progress.pdf)



준비 과정에서 고려하였으나, 다음과 같은 내용을 다루지 못하였다.

- `\frontmatter`와 `\tableofcontents` 등 TOC.
- `\makeindex`와 `\index`, 그리고  $\langle theindex \rangle$  (색인)
- `chaptertoc`
- source code listing

- (1) 언젠가 빠뜨린 내용을 보충하여 더 확장된 문서 디자인 공부를 할 기회가 있으면 좋겠다.
- (2) 필요한 거의 대부분을 배운 셈이다. *memoir manual*과 좋은 L<sup>A</sup>T<sub>E</sub>X 문헌들이 가까이 있고 KTUG 질답게시판이 아군임을 믿고, 차근차근 원하는 결과를 얻어가기를 바란다.
- (3) 형식과 디자인, 그리고 모양의 화려함이 내용의 빈곤함과 만나면 누더기에 불과하다. 오로지 훌륭한 내용을 잘 전달하기 위한 도구일 때 의미가 있을 따름이다. 더 좋은 논문, 더 좋은 책을 쓸 수 있도록 더 열심히 공부하자.
- (4) L<sup>A</sup>T<sub>E</sub>X은 “문서를 프로그래밍”하는 것이다. 프로그래밍은 논리적 추론 과정의 소산이다. 이 점을 명심하면 결코 어렵지 않다(고 생각한다).