

## LECTURE 1. 알아두어야 할 기본 사항

**1.1 라텍 파일 형식** 라텍이 처리하는 파일의 형식은 텍스트 파일이다. 이 파일 형식은 빔(Vim)이나 이맥스(Emacs) 같은 대표적인 텍스트 편집기는 물론 마이크로소프트 윈도의 보조 프로그램인 메모장(Notepad)으로도 만들 수 있다. 텍스트 파일의 크기는 다른 파일 형식에 비해 매우 작으며, 설정 파일에 예기치 못한 손상이 발생할지라도 파일 전체를 읽지 못하는 일은 발생하지 않는다.  $\TeX$  시스템에서는 TeXworks라는 통합환경을 이용해 라텍 파일을 만든다.

**1.2 라텍 파일 템플릿** 라텍 파일에는 문서의 내용과 논리 구조, 그리고 조판 방법을 알려주는 지시사항들이 들어간다. 이러한 구성 요소들은 일정한 형식에 맞게 작성되어야 한다. 앞으로  $\TeX$  시스템에서 사용하게 될 라텍 파일의 기본 템플릿(template)은 다음과 같다.

<code>%!TEX TS-program = xelatex</code>	<code>%!TEX TS-program = xelatex</code>
<code>%!TEX encoding = UTF-8 Unicode</code>	<code>%!TEX encoding = UTF-8 Unicode</code>
<code>\documentclass[a4paper]{memoir}</code>	<code>\documentclass[a4paper]{oblivoir}</code>
<code>\usepackage{xltextra}</code>	
<code>\begin{document}</code>	<code>\begin{document}</code>
Small is beautiful.	작은 것은 아름답다.
<code>\end{document}</code>	<code>\end{document}</code>

영어 문서를 작성하는 경우에는 왼쪽 템플릿을, 한글 문서를 작성하는 경우에는 오른쪽 템플릿을 사용한다. 두 가지 템플릿 모두 첫 두 줄은 ‘%!TEX’으로 시작하는데, 이들은 TeXworks 통합환경에게 조판 방법을 알려주는 지시사항들이다.

**1.3 전처리부와 주요부** 라텍 파일은 `\documentclass[...]{...}` 명령으로 시작하는 것이 일반적이다. 이 명령은 문서의 기본 형태를 결정해 주는 클래스 파일을 부른다. 예를 들어, [1.2] 라텍 파일 템플릿에서 영어 문서를 위한 템플릿의 경우 memoir 클래스를, 그리고 한글 문서를 위한 템플릿의 경우 oblivoir 클래스를 사용하고 있다. 하지만 oblivoir 클래스가 내부에서 memoir 클래스를 불러들이므로 문서의 기본 형태를 정하는 방법에는 별 차이가 없다.

클래스 파일을 부른 다음 필요하다면 `\usepackage{...}` 명령으로 패키지 파일을 부를 수 있다. 패키지 파일에는 라텍에 들어있는 기본 논리 구조이외에 다른 논리 구조들을 추가하는 명령 또는 논리 구조의 모양(스타일)을 변경하는 명령들이 들어있다.

이러한 설정 과정이 모두 끝나면 `\begin{document}` 명령과 함께 문서의 **주요부**가 시작된다. 특별히 `\documentclass[...]{...}`와 `\begin{document}` 사이에 오는 부분을 **전처리부(preamble)**라 부른다. 문서의 내용은 주요부에서 논리 구조를 주는 라텍 명령들과 함께 입력된다. 문서의 끝은 `\end{document}` 명령으로 끝난다. 이 명령 이후에 나오는 모든 것은 무시된다.

**1.4 공백문자** 라텍은 빈 칸, 탭 등과 같은 공백(whitespace)문자들을 모두 동일하게 “스페이스(space)”로 처리한다. 이어지는 여러 개의 공백문자들은 하나의 스페이스로 취급한다. 행의 첫머리에 있는 공백문자들은 무시하고 한 번의 줄바꿈 역시 공백문자로 간주한다.

두 줄 사이에 빈 줄을 하나 넣으면 기존 문단이 끝나고 새로운 문단이 시작된다. 빈 줄을 여러 개 넣어도 한 줄을 넣은 것과 효과는 같다. 다음 예에서 왼쪽은 라텍 입력 파일의 주요부, 즉 `\begin{document}`와 `\end{document}` 사이에 들어가는 내용이고, 오른쪽은 출력된 결과물이다.

```
It does not matter whether you enter
one or several      spaces after a word.

An empty line starts a new paragraph.
```

```
It does not matter whether you enter one or several
spaces after a word.

An empty line starts a new paragraph.
```

**1.5 특수문자** 다음 기호들은 라텍에서 특별한 의미를 가지므로 주의해서 사용해야 한다. 만일 이러한 문자들을 문서에서 표시하려면 반드시 백슬래시(\)를 앞에 붙여 주어야 한다.

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

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

백슬래시 문자(\)는 백슬래시를 그 앞에 하나 더 붙인다고 얻어지지 않는다. 대신 이 명령(\\)은 강제 줄바꿈에 사용된다. 백슬래시 문자는 `\textbackslash` 명령으로 표시할 수 있다.

**1.6 주석** 앞의 1.5 특수문자들 중 퍼센트 기호(%)는 특별한 용도로 쓰인다. 라텍은 입력 파일을 처리하다가 이 문자를 만나면 그 줄의 나머지 부분을 모두 무시한다. 또한 그 다음 줄 첫머리에 오는 1.4 공백문자들도 모두 무시한다. 컴퓨터 프로그래밍에서 주석(comment)을 다는 것과 같이 라텍 입력 파일에도 특수문자 %를 이용해 주석을 넣을 수 있다. 주석은 당연히 최종 출력물에 나타나지 않는다.

```
This is an % stupid
% Better: instructive <----
example: Supercal%
         ifragilist%
         icexpialidocious
```

```
This is an example: Supercalifragilisticexpialidocious
```

**1.7 라텍 명령** 라텍 명령은 대소문자를 구별하며 다음 두 가지 가운데 하나의 형태를 가진다.

- 백슬래시(\)로 시작하여 문자(letter)만으로 이루어진 이름을 갖는다. 명령 이름은 1.4 공백문자나 숫자 또는 '문자 아닌 것'이 오면 끝난다.
- 백슬래시 다음에 딱 하나의 1.5 특수문자가 온다.

라텍 명령 다음의 1.4 공백문자들은 무시된다. 그렇지 않으려면 다음 예와 같이 라텍 명령 다음에 {} 또는 백슬래시(\)를 붙이면 된다.

```
Today is \today. I read that Knuth divides
the people working with \TeX{} into
\TeX{}nicians and \TeX perts.
```

```
Today is November 6, 2010. I read that Knuth divides the
people working with TEX into TEXnicians and TEXperts.
```

인자(parameter)가 필요한 명령의 경우 라텍 명령 다음에 오는 중괄호 {...} 안에 인자를 넣는다. 옵션인자(optional parameter)를 쓰는 라텍 명령의 경우 대괄호 [...] 안에 옵션인자를 넣는다.

```
You can \textsl{lean} on me!
\makebox[Opt][l]{Duplicate me!}%
Duplicate me!
```

You can *lean* on me! Duplicate me!

**1.8 마침표와 약어** 라텍은 단어 사이의 간격을 적절히 조정해서 문단의 양끝을 가지런히 정렬한다. 또한 글을 읽기 쉽게 하기 위해 각 문장의 끝에는 공백을 약간 더 삽입한다. 라텍은 마침표 다음에 [1.4] 공백문자가 오는 경우 문장의 끝으로 인식한다. 하지만 대문자 다음에 바로 마침표가 오는 경우 약어일 경우가 많으므로 문장의 끝으로 여기지 않는다.

이러한 두 가지 규칙을 벗어나는 경우에는 마침표가 문장의 끝을 의미하는지 그렇지 않은지 사용자가 직접 지정해 주어야 한다. 마침표 다음에 공백문자가 오지만 문장의 끝이 아닌 경우에는, 공백문자 앞에 백슬래시 (\)를 넣거나 공백문자 대신 틸데(~)를 쓰면 된다. 또한 대문자 다음에 마침표가 오지만 문장의 끝인 경우에는, 마침표 앞에 \@ 명령을 붙여주면 된다.

```
Mr.-Smith was happy to see her \
cf.-Fig.-5 \
I like BASIC\@. What about you?
```

Mr. Smith was happy to see her  
cf. Fig. 5  
I like BASIC. What about you?

**1.9 따옴표** 라텍에서는 동일한 따옴표 문자(")를 여는 따옴표와 닫는 따옴표 모두에 사용하지 않는다. 대신 여는 따옴표에는 두 개의 그라베 악센트(grave accent; `), 그리고 닫는 따옴표에는 두 개의 작은 따옴표(vertical quote; ')를 사용한다. 따옴표 대신 작은 따옴표를 쓸 경우에는 각각의 기호를 한 번씩만 사용하면 된다.

```
` `Please press the `x' key.'`
```

"Please press the 'x' key."

따옴표와 작은 따옴표가 연이어 나오는 경우, 이들 사이에 \, 명령으로 미세한 간격을 주면 좋다.

```
`\, `Beginning' and `End',`
```

" `Beginning' and `End' "

**1.10 대시와 하이픈** 문서를 작성할 때에는 네 종류의 대시 모양을 용도에 맞게 구별해서 사용해야 한다. 라텍은 입력하는 대시문자(-)의 수에 따라 다른 모양을 주는데, 각각 하이픈(hyphen; -), 엔대시(en dash; –), 엠대시(em dash; —)라 부른다. 다른 한 가지 대시는 수학에서 쓰는 빼기(minus; −) 부호인데 수학모드에서 자동으로 표시된다. 이들 대시들의 용도는 다음 예와 같다.

```
daughter-in-law, X-rated \
pages 13--67 \
his attendants---all dressed in white---were
grouped around him.
yes---or no? \
$0$, $1$, and $-1$
```

daughter-in-law, X-rated  
pages 13–67  
his attendants—all dressed in white—were grouped  
around him. yes—or no?  
0, 1, and −1

**1.11 과제** Project Gutenberg (<http://www.gutenberg.org>) 홈페이지에서 Herman Melville (1819–1891)이 쓴 소설 “모비딕 (Moby Dick)”을 텍스트 파일 (Plain Text UTF-8) 형식으로 내려받은 후, 첫째 장 (chapter)의 처음 네 문단을 다음 라텍 템플릿을 이용해 출력하라.

---

```
%!TEX TS-program = xelatex
%!TEX encoding = UTF-8 Unicode

\documentclass[a4paper]{memoir}
\usepackage{xltextra}

\begin{document}

\title{Moby Dick}
\author{Herman Melville}
\maketitle

Call me Ishmael. Some years ago---never mind how long precisely---having
little or no money in my purse, and nothing particular to interest me on
shore, I thought I would sail about a little and see the watery part of
... (이하 생략) ...

\end{document}
```

---

## LECTURE 2. 문서 구조와 레이아웃

**2.1 클래스와 옵션** 앞으로 사용할 기본 클래스 파일은 [1.2] 라텍 파일 템플릿에서 보았듯이 영어 문서의 경우 memoir 클래스, 그리고 한글 문서의 경우 memoir 클래스를 기반으로 한 oblivoir 클래스이다.

영어 문서를 위한 [1.2] 라텍 파일 템플릿에서 memoir 클래스를 부를 때 다음 명령과 같이 a4paper 옵션이 함께 주어졌다. 이 옵션은 출력용지(stock paper)로 A4(297 × 210mm)를 설정하는데, 이 옵션이 주어지지 않으면 미국에서 주로 사용하는 U.S. 레터(11 × 8.5in)가 선택되므로 주의해야 한다.

```
\documentclass[a4paper]{memoir}
```

클래스 옵션은 a4paper 하나만 주어졌지만 memoir 클래스는 기본값으로 다음과 같은 클래스 옵션들을 함께 지정한다.

```
\documentclass[a4paper,10pt,twoside,onecolumn,openright,final,portrait]{memoir}
```

먼저 10pt는 문서의 기본 글꼴 크기를 설정하는 옵션이다. 이외에도 memoir 클래스는 9pt, 11pt, 12pt, 14pt, 17pt 등과 같은 다양한 크기의 옵션들을 제공한다. 하지만 9pt, 10pt, 11pt, 12pt 옵션을 제외한 다른 크기의 옵션들은 잘 쓰이지 않는다.

옵션 twoside는 최종 출력물이 양면 인쇄용임을 라텍에게 알려준다. 이 옵션을 쓰면 제본을 위한 여백까지 고려하므로 홀수 쪽과 짝수 쪽의 여백의 크기가 달라지게 된다. 제본을 염두에 둔 책이나 보고서 등을 제작할 때 이 옵션을 주로 쓴다. 굳이 제본이 필요없는 프린터 인쇄물을 만들거나 홀수 쪽과 짝수 쪽의 여백의 크기가 동일하길 원한다면 대신 oneseide 옵션을 사용하도록 한다.

하나의 판면이 두 개의 컬럼으로 나뉘어진 문서를 작성할 경우에는 onecolumn 옵션 대신 twocolumn 옵션을 쓴다. 새로운 장(chapter)이 시작하는 쪽의 위치는 “항상 오른쪽(openright)” 대신 “항상 왼쪽(openleft)” 또는 “아무 쪽이나(openany)”로 바꿀 수 있다.

옵션 final 대신 draft를 쓰면 약간의 조판 정보를 더 보여주지만 최종 출력물에는 적합하지 않다. 끝으로 portrait 옵션 대신 landscape을 쓰면 출력용지의 가로 크기와 세로 크기가 서로 바뀐다.

**2.2 여백 설정** 라텍에서 판면의 구성을 바꾸는 일은 그리 간단하지 않다. 좌우 여백을 조정한 후 판면을 중앙에 위치하게 만드는 가장 기본적인 일조차 memoir 클래스 대신 다른 클래스 파일을 사용한다면 쉽지 않다. 판면의 구성을 위해 memoir 클래스가 다양한 방법을 제공하고 있지만 이와 관련된 여러 가지 용어들과 명령들을 모두 익히는 것은 버거운 것이 사실이며, 굳이 그렇게 할 필요도 없다. 대신 좌우 및 상하 여백을 조절함으로써 인쇄 영역의 크기를 조절하는 간단한 방법을 소개한다.

```
\setlrmarginsandblock{1in}{1in}{*}
\setulmarginsandblock{1in}{1in}{*}
\checkandfixthelayout
```

위의 명령들은 반드시 라텍 입력 파일의 전처리부(preamble), 즉 \begin{document} 명령이 시작되기 전에 넣어야 한다. 첫째 줄은 판면의 좌우 여백을 설정하는데 쓰인다. 세 개의 인자들 중 처음 두 가지 인자들은

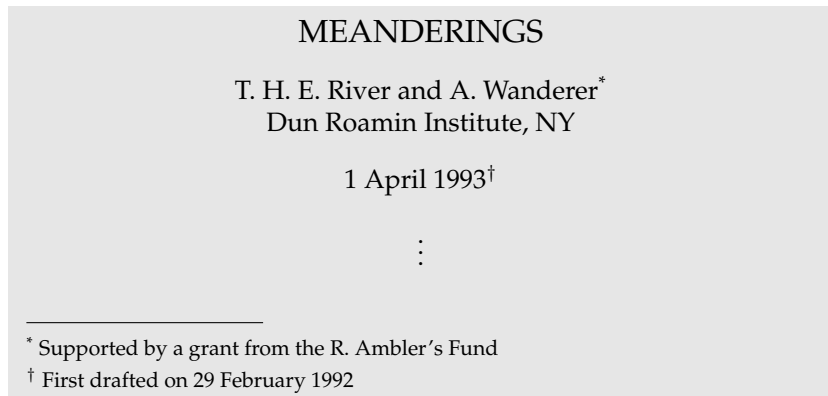
각각 좌측 여백과 우측 여백의 폭을 의미한다. 마찬가지로 방법으로 판면의 상하 여백도 둘째 줄과 같은 방법으로 설정할 수 있다.

문서의 가독성을 해치지 않으려면 가급적 좌우 및 상하 여백을 1인치 이상 유지하도록 한다. 특히 좌우 여백이 좁은 경우 인쇄 영역의 가로 폭이 한 눈에 들어오지 않아 가독성이 현저히 떨어지게 된다.

**2.3 표제** 표제에는 문서의 제목 및 저자에 관련된 사항들이 들어간다. 표제를 출력하고자 하는 경우 라텍 입력 파일의 주요부에 다음과 같이 내용을 입력한다.

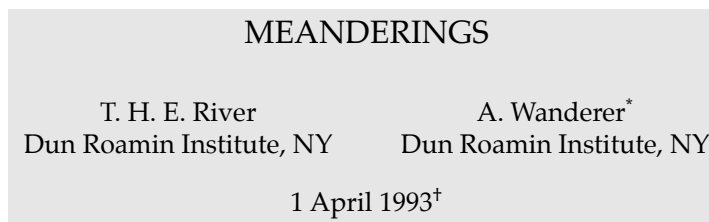
```
\title{MEANDERINGS}
\author{T. H. E. River and A. Wanderer\thanks{Supported by ... Fund} \
      Dun Roamin Institute, NY}
\date{1 April 1993\thanks{First drafted on 29 February 1992}}
\maketitle
```

출력된 결과물은 다음 그림과 같다. 위에서 \thanks{...} 명령으로 입력한 내용은 [2.6] 각주로 표시된다.

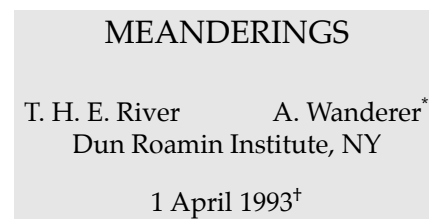


위의 예와 같이 “and” 라는 단어를 직접 넣는 대신 아래와 같이 \and 명령을 사용하면 저자들을 적당한 공백으로 나누는 효과를 얻는다. 또한 \and 대신 \andnext 명령을 사용하면 공백 대신 줄이 바뀌게 된다.

```
\author{
  T. H. E. River \
  Dun Roamin Institute, NY
\and
  A. Wanderer\thanks{...} \
  Dun Roamin Institute, NY
}
```



**연습문제** 저자들의 소속이 같은 경우 위의 그림 대신 오른쪽 그림처럼 소속을 한 번만 나타내는 방법도 있다. 어떤 명령을 쓰면 될까? (힌트: \and와 \andnext를 이용하라.)



**2.4 장절 명령** 문서에 논리 구조를 주는 방법들 중 대표적인 것이 장(chapter), 절(section), 소절(subsection) 등으로 문서를 잘 나누는 것이다. 이렇게 구성된 문서는 조판에 드는 시간과 수고를 줄여줄 뿐만 아니라 출력된 결과물의 가독성도 높여준다.

memoir 클래스가 제공하는 장절 명령들은 다음과 같다. 이들 명령들은 장절 제목을 인자로 가진다. 기본 출력 형태는 오른쪽 그림과 같다.

```
\chapter{Introduction}
\section{Section}
\subsection{Subsection}
\subsubsection{Subsubsection}
```

그리 자주 사용되지는 않지만 하위 레벨의 명령으로 `\paragraph{...}`과 `\subparagraph{...}`도 있다.

이들 여섯 가지 장절 명령들은 모두 자동으로 번호를 매긴다. 번호가 매겨지지 않는 장절 제목으로 바꾸려면 명령과 인자 사이에 \*를 붙이면 된다. 예를 들어, `\section{Section}` 대신 `\section*{Section}`을 사용하면 절의 제목만 나타나고 번호는 사라진다.

만약 `\chapter{...}` 명령을 부르지 않은 상태에서 `\section{...}` 명령을 사용하면 절 번호가 '0.1'로 나타난다. 문서에서 “장”을 전혀 사용하지 않거나, 또는 “장”을 사용하더라도 장 번호를 절 번호 앞에 표시하지 않으려면 다음 명령을 라텍 파일의 [1.3] 전처리부에 넣도록 한다.

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

첫째 줄은 절 번호에 장 번호가 나타나지 않도록 막아 주는 명령이다. 두 번째 줄은 새로운 장이 시작될 때마다 절 번호도 처음부터 다시 시작하도록 만들어 주는 명령이므로 “장”을 사용하지 않는 문서라면 필요하지 않다.

장절 명령 `\subsection`과 `\subsubsection` 모두 자동으로 번호를 매기지만 위의 그림에서 번호가 표시되지 않았다. 번호를 표시하고 싶다면 다음 명령과 같이 번호가 표시되는 장절 명령의 레벨을 바꾸면 된다.

```
\setsecnumdepth{subsubsection}
```

**2.5 차례** 라텍은 문서를 컴파일하면서 장절 제목과 쪽 번호를 외부 파일에 기록해 둔다. `\tableofcontents` 명령은 이 명령이 위치한 곳에 외부 파일에 기록된 정보를 이용해 차례(contents)를 만들어 넣는다. 올바른 차례를 얻기 위해 문서를 두 번 또는 세 번 컴파일하는 경우도 있다.

번호가 표시되는 [2.4] 장절 명령의 레벨을 바꾸는 `\setsecnumdepth{...}` 명령처럼 차례에 표시되는 장절 명령의 레벨도 다음 명령과 같이 바꿀 수 있다.

```
\settocdepth{subsubsection}
```

## Chapter 1

### Introduction

#### 1.1 Section

##### Subsection

##### Subsubsection

**2.6 각주** 각주가 참조하는 낱말이나 문장 바로 뒤에 (공백 없이) `\footnote{...}` 명령을 두면 각주가 현재 쪽의 하단에 만들어 진다. 한 문장, 혹은 문장의 일부를 참조하는 각주는 쉼표나 마침표 뒤에 놓아야 한다.

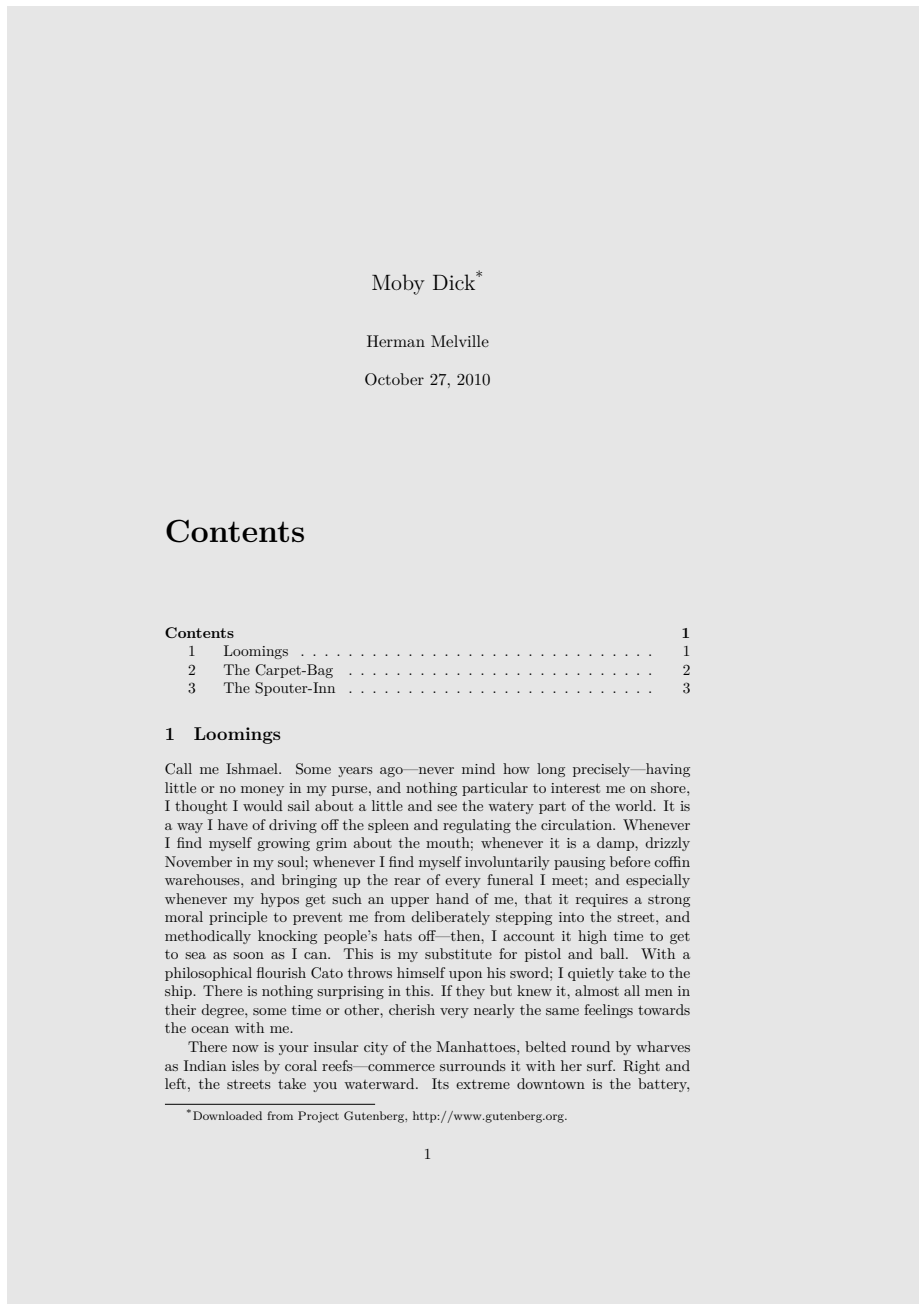
```
Footnotes\footnote{This is
  a footnote.} are often used
by people using \LaTeX.
```

Footnotes<sup>a</sup> are often used by people using  $\LaTeX$ .

---

<sup>a</sup>This is a footnote.

**2.7 과제** [2.7] 과제에서 다운로드한 소설 “모비딕 (Moby Dick)”의 처음 세 개의 장(chapter)에 대해, 각 장의 처음 네 문단을 조판하라. 이 때 출력물의 첫 페이지는 다음 그림과 같이 나오도록 한다.





### LECTURE 3. 텍스트와 글꼴

**3.1 강조** 타자기로 작성된 문서에서 단어나 문장을 강조하기 위해 밑줄을 그은 모습을 자주 볼 수 있다. 라텍에서 `\underline{...}` 명령을 이용해 밑줄을 그을 수 있지만 이 방법을 인쇄물에 사용하는 것은 권하지 않는다.

영어 인쇄물에는 강조하고자 하는 단어나 문장의 글자 모양을 ‘이탤릭 (italic)’으로 바꾸는 강조 방법이 주로 쓰인다. 라텍도 같은 강조 방법을 쓴다. 강조할 내용이 짧은 경우 `\emph{...}` 명령을 사용하고, 그렇지 않은 경우 같은 결과를 주는 `{\em ...}` 명령을 사용한다. 강조된 문장 안에서 다시 강조된 단어(다음 예에서 ‘normal’)나 문장은 이탤릭이 아니라 원래의 글자 모양을 가진다.

```
{\em If you use emphasizing inside a piece
of emphasized text, then \LaTeX{} uses the
\emph{normal} font for emphasizing.}
```

*If you use emphasizing inside a piece of emphasized text, then  $\LaTeX$  uses the normal font for emphasizing.*

일본의 조판 방법을 많은 부분 관행으로 따르고 있는 한글 인쇄물의 경우 이탤릭 대신 고딕 혹은 돋움 계열로 글꼴 자체를 바꾸는 방법을 사용해 왔다. 이러한 관행을 탈피하려는 시도로 최근에는 ‘드러냄표’를 이용한 강조 방법도 쓰고 있다. 이것은 강조하고자 하는 어구에 점을 찍는 것이다. 한글 문서를 위한 1.2 라텍 파일 템플릿에서와 같이 `oblivoir` 클래스를 부르면 `\dotemph{...}` 명령으로 드러냄표를 쓸 수 있다.

가로쓰기에서는 `\dotemph{점을 글자 위에 찍으므로}` 傍점이 아니라 上점이 맞겠지만 관행적으로 방점이라 불러왔다.

가로쓰기에서는 점을 글자 위에 찍으므로 傍점이 아니라 上점이 맞겠지만 관행적으로 방점이라 불러왔다.

**3.2 글자 크기** 조판에서 문서의 글자 크기는 절대적 크기 또는 상대적 크기로 주어진다.

절대적 크기는 워드프로세서에서 주로 쓰인다. 툴바에 위치한 글자 크기 조절 창에서 여러 숫자들 중 하나를 선택하거나 직접 숫자를 입력할 때 사용된다. 글자의 크기를 바로 확인할 수 있다는 장점이 있지만 문서의 논리 구조는 완전히 무시되는 단점도 있다. 이렇게 만들어진 문서의 글자들을 모두 같은 비율로 키우거나 줄여야 한다고 상상해 보라. 여기에 드는 엄청난 시간과 수고는 누구라도 피하려 할 것이다.

논리 구조를 가진 문서에서 글자 크기는 상대적 크기, 즉 기본 글자 크기에 대한 상대값으로 주어진다. 기본 글자 크기보다 조금 큰 값, 큰 값, 많이 큰 값과 같은 식이다. 문서의 글자들을 모두 같은 비율로 키우거나 줄여야 하는 경우라도 기본 글자 크기만 바꿈으로써 해결할 수 있다.

라텍에서는 절대적 크기와 상대적 크기를 함께 쓸 수 있지만 문서의 논리 구조를 해치지 않기 위해 상대적 크기를 주로 쓴다. 다음 예에서 상대적 글자 크기를 지정하는 열 가지 라텍 명령들을 볼 수 있다. 기본 글자 크기를 주는 명령은 `\normalsize`이다. memoir 클래스에는 이외에도 가장 작은 크기의 `\miniscule` 명령과 가장 큰 크기의 `\HUGE` 명령이 있다.

```
{\tiny Smallest} {\scriptsize Very small}
{\footnotesize Smaller} {\small Small}

{\normalsize Normal} {\large Large}
{\Large Larger} {\LARGE Even larger}

{\huge Still larger} {\Huge Largest}
```

Smallest Very small Smaller Small  
 Normal Large Larger Even larger  
 Still larger Largest

**3.3 글꼴 속성** 글꼴 혹은 폰트를 다루는 일은 조판에서 매우 중요하다. 제대로 역할을 했을 때와 그렇지 못했을 때 모두 확연히 두드러진 결과가 나타난다는 점에서, 야구의 투수 또는 축구의 스트라이커에 비길 수 있다.

글꼴을 다루는 방법은 문서의 종류와 목적에 따라 달라져야 한다. 포스터나 책 표지에는 사람들의 주목을 끌기 위해 다양한 변화를 준 글꼴들을 많이 사용한다. 반면에 글이 판면의 대부분을 차지하는 문서라면 화려한 글꼴들이 도리어 가독성을 떨어뜨린다. 라텍은 후자의 경우를 위해 만들어진 조판 도구이다. 전자를 위한 조판 도구와 비교하는 것은 태블릿과 노트북을 비교하는 것과 마찬가지로이다.

라텍은 글꼴 패밀리, 모양, 그리고 굵기의 세 가지 글꼴 속성을 이용해 글꼴을 다룬다. 절대적인 글자 크기 대신 상대적인 크기를 주로 사용하는 것과 마찬가지로 글꼴도 직접 글꼴 파일을 지정하는 방법 대신 글꼴 속성을 이용한 상대적 설정 방법을 선호한다. 이들 속성들은 다음 예가 보여주듯 서로 독립적으로 작동한다. 즉, 한 속성의 변화가 다른 두 속성에 영향을 미치지 않는다.

```
Mixing {\bfseries different series,
\textsf{families}}
and \textsl{\texttt{shapes,}}
{\scshape especially in one sentence,}
is usually \emph{highly inadvisable!}

normal and {\bfseries bold and
\slshape slanted} and back} again.
```

Mixing **different series, families** and *shapes*, ESPECIALLY IN ONE SENTENCE, is usually **highly inadvisable!**  
normal and **bold and slanted and back** again.

세 가지 글꼴 속성을 모두 기본값으로 되돌리는 명령은 `{\normalfont ...}`이다. 내용이 짧을 경우 `\textnormal{...}`을 대신 쓸 수 있다. 이제 각 속성들에 대해서 살펴보자.

**3.4 글꼴 패밀리** 라텍은 기본으로 세 가지 글꼴 패밀리를 사용한다. 로마 패밀리(`\rmfamily`)에 대응되는 글꼴은 본문에 주로 쓰이는 세리프 글꼴이다. 획의 시작과 끝 부분에 있는 작은 돌출선을 '세리프(serif)'라 부르는데 '산세리프(sans serif)'는 세리프가 없다는 뜻이다. 산세리프 패밀리(`\sffamily`)에는 이름 그대로 산세리프 글꼴이 대응되는데 프레젠테이션에 많이 사용된다. 모든 글자들의 폭이 일정한 모노스페이스(monospace) 글꼴이 대응되는 타이프라이터 패밀리(`\ttfamily`)는 프로그램 소스를 출력하는 경우에 한해 한정적으로 사용된다.

```
{\rmfamily to switch to a Roman font,} \\  
{\sffamily to select a sans serif font,} \\  
{\ttfamily to switch to a typewriter font.}
```

to switch to a Roman font,  
to select a sans serif font,  
to switch to a typewriter font.

내용이 짧은 경우 `\textrm{...}`, `\textsf{...}`, 그리고 `\texttt{...}` 명령을 각각 사용할 수 있다. `oblivoir` 클래스에서 한글을 사용하는 경우, 로마 패밀리에는 바탕 또는 명조 계열의 글꼴이 대응되고, 산세리프 패밀리에는 돋움 또는 고딕 계열의 글꼴이 대응된다. 타이프라이터 패밀리에는 손글씨 또는 고딕 계열의 글꼴들 중 폭이 일정한 글꼴이 대응된다.

```
\textrm{바탕 또는 명조} \\  
\textsf{돋움 또는 고딕} \\  
\texttt{폭이 일정한 손글씨 또는 고딕}

{\sffamily 폭이 다른 돋움 또는 고딕}과 \\  
{\ttfamily 폭이 같은 돋움 또는 고딕}의 차이
```

바탕 또는 명조  
돋움 또는 고딕  
폭이 일정한 손글씨 또는 고딕  
폭이 다른 돋움 또는 고딕과  
폭이 같은 돋움 또는 고딕의 차이

**3.5 글꼴 모양** 라텍이 제공하는 글꼴 모양(shape)은 네 가지가 있다. 라틴 글꼴의 대표적인 모양 변화는 ‘이탤릭’으로 `\itshape` 명령을 이용해 바꿀 수 있다. 이탤릭과 비슷해서 혼동하기 쉬운 ‘기울어진(slanted)’ 모양은 `\slshape` 명령으로 바꾼다. 이탤릭은 독립된 글꼴 파일이 주어지는데 반해 기울어진 모양은 기본 모양과 동일한 글꼴 파일을 사용한다. 라틴 글꼴의 또 다른 모양 변화인 ‘작은 대문자(small caps)’ 모양은 `\scshape` 명령으로 바꾼다. 작은 대문자 모양을 위한 글자들은 기울어진 모양과 달리 기본 모양의 글꼴 파일 안에 독립된 글자 형태로 들어있는 경우가 많다.

아무런 변화가 없는 기본 모양은 `\upshape` 명령으로 지정한다. 지정된 모양에 맞는 글꼴이 없는 경우 다른 모양으로 대체된다. 글꼴 패밀리와 마찬가지로 내용이 짧은 경우 `\textup{...}`, `\textit{...}`, `\textsl{...}`, 그리고 `\textsc{...}` 명령을 각각 사용할 수 있다.

```
{\upshape to switch to an upright font,} \\  
{\itshape to select an italic shape,} \\  
{\slshape to choose a font that is slanted,} \\  
{\scshape to switch to Caps and Small Caps.}
```

```
to switch to an upright font,  
to select an italic shape,  
to choose a font that is slanted,  
TO SWITCH TO CAPS AND SMALL CAPS.
```

한글은 물론 한자를 중심으로 한 동아시아 문자들의 경우 글꼴의 모양 변화는 거의 사용되지 않는다. 간혹 기울어진 모양을 포함한 문서도 있지만 동아시아 문자들의 타이포그래피를 전혀 고려하지 않고 라틴 문자들의 타이포그래피를 맹목적으로 답습한 것에 다름 아니므로 결코 바람직한 선택이 아니다.

**3.6 글꼴 굵기** 글자의 굵기(weight) 변화는 숫자 또는 이름으로 주어지는데, 라텍에서는 이름만 사용한다. 가장 대표적인 것은 ‘볼드(bold)’로 글자 형태는 그대로인 채 획의 굵기만 진하게 만든다. 라텍은 기본 굵기를 주는 `\mdseries` 명령 이외에 볼드를 주는 `\bfseries` 명령만 제공한다. 마찬가지로 내용이 짧은 경우 `\textmd{...}` 와 `\textbf{...}` 명령을 각각 사용할 수 있다.

```
{\mdseries to switch to medium weight,} \\  
{\bfseries to select a boldface font.}
```

```
to switch to medium weight,  
to select a boldface font.
```

이탤릭을 위한 독립된 글꼴 파일이 있듯이 볼드도 독립된 글꼴 파일로 주어진다. 하지만 자체 글꼴 디자인이 없으면 흉내조차 낼 수 없는 이탤릭에 반해 볼드는 자체 디자인이 없더라도 같은 글자를 여러 번 식자함으로써 비슷한 효과를 낼 수 있다. 이 방법은 최종 출력물의 품위를 낮추므로 피치 못할 경우를 제외하고는 사용하지 않는 것이 좋다. 즉, 볼드를 위한 글꼴 파일이 따로 없는 경우 볼드 사용을 가급적 자제하라는 뜻이다.

**3.7 글꼴 변경**  $\TeX$ 에서 글꼴 파일들을 글꼴 패밀리에 대응시키는 방법은 다음 예와 같다. 로만 패밀리, 산 세리프 패밀리, 타이프라이터 패밀리에 해당하는 명령은 각각 `\setmainfont`, `\setsansfont`, `\setmonofont` 이다. 이 명령들의 인자에는 주로 운영체제에서 사용하는 글꼴의 이름을 넣는다. 또한 이 명령들은 [1.3] 전처리부에 두는 것이 일반적이다.

```
\setmainfont[Mapping=tex-text]{Times New Roman}  
\setsansfont[Mapping=tex-text]{Arial}  
\setmonofont{Courier New}  
  
{\rmfamily Times New Roman for a Roman font} \\  
{\sffamily Arial for a sans serif font} \\  
{\ttfamily Courier New for a monospace font}
```

```
Times New Roman for a Roman font  
Arial for a sans serif font  
Courier New for a monospace font
```

oblivoir 클래스를 사용하는 경우, 세 가지 글꼴 패밀리에 한글 글꼴 파일을 대응시키는 방법은 다음 예와 같다. 한글 뿐만 아니라 한자도 `\setmainhanjafont`, `\setsanshanjafont`, `\setmonohanjafont` 명령을 이용해 다른 글꼴로 대체할 수 있다. 한자가 함께 들어있는 한글 글꼴은 많지 않다는 사실을 기억하라. 한글 글꼴에 비해 한자 글꼴을 만드는데 시간과 비용이 많이 들기 때문이다.

```
\setmainhangulfont[Mapping=tex-text]{Batang}
\setsanshangulfont[Mapping=tex-text]{Dotum}
\setmonohangulfont{NanumGothicCoding}
```

```
{\rmfamily 로만 글꼴은 바탕} \
{\sffamily 산세리프 글꼴은 돋움} \
{\ttfamily 모노 글꼴은 나눔고딕코딩}
```

로만 글꼴은 바탕  
산세리프 글꼴은 돋움  
모노 글꼴은 나눔고딕코딩

문서의 논리 구조를 해치기 때문에 바람직하지 않지만 글꼴을 잠시 변경할 일이 있을 때, 라틴 글꼴의 경우 `\fontspec` 명령, 그리고 한글 글꼴의 경우 `\hangulfontspec` 명령을 사용한다. 이 명령들은 앞의 예제들에서 사용한 글꼴 변경 명령들과 동일한 인자 및 옵션인자를 받아들인다.

```
{\fontspec{Minion Pro} Minion Pro} and
{\fontspec{Myriad Pro} Myriad Pro}
```

```
{\hangulfontspec{Malgun Gothic} 맑은 고딕}과
{\hangulfontspec{Gungsuh} 궁서}
```

Minion Pro and Myriad Pro  
맑은 고딕과 궁서

글꼴 변경 명령에 사용되는 옵션인자들은 그 수도 너무 많고 내용도 복잡하다. 위의 예들에서 자주 사용된 ‘Mapping=tex-text’만 살펴보자. 이 옵션은 [1.10]의 엔대시(-), 엠대시(-), 그리고 [1.9]따옴표 등을 적절한 글자들로 대체하는 역할을 한다. 하지만 프로그램 소스를 출력하는데 주로 쓰이는 타이프라이터 패밀리에 이 옵션을 주는 것은 바람직하지 않다.

```
{\fontspec{Georgia}
`endash' -- ``emdash" ---}
{\fontspec[Mapping=tex-text]{Georgia}
`endash' -- ``emdash'' ---}
```

`endash' -- ``emdash" --- ‘endash’ – “emdash” –

**3.8 과제** [2.7]과제에서 만든 문서의 본문 글꼴을 ‘Times New Roman’, ‘Minion Pro’, 그리고 ‘Palatino Linotype’으로 바꾸어 조판한 후 그 결과를 비교하라. 만약 ‘Palatino Linotype’가 없는 경우 본문 글꼴로 가장 적당하다고 생각하는 글꼴을 대신 선택하라.

## LECTURE 4. 텍스트와 환경

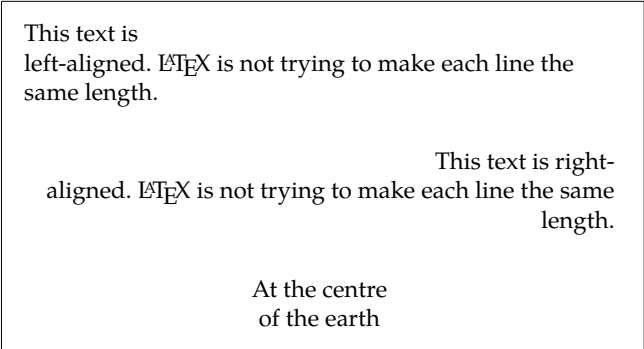
라텍 환경 (environment)이란 `\begin{환경이름}` 명령으로 시작해서 `\end{환경이름}` 명령으로 끝나는 것을 일컫는다. 일반적으로 [1.7](#) 라텍 명령과 구별해서 사용한다. 환경은 문서에 논리구조를 주는 대표적인 방법이다.

**4.1 문단 정렬** 라텍은 기본으로 세 가지 문단 정렬 방법을 제공한다. 왼쪽 정렬은 `flushleft` 환경, 오른쪽 정렬은 `flushright` 환경, 그리고 가운데 정렬은 `center` 환경으로 주어진다. 문단은 정해진 방법으로 자동 정렬되는데 강제 줄바꿈이 필요한 경우 `\\` 명령을 사용하면 된다.

```
\begin{flushleft}
This text is\\ left-aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushleft}

\begin{flushright}
This text is right-\\aligned.
\LaTeX{} is not trying to make
each line the same length.
\end{flushright}

\begin{center}
At the centre\\of the earth
\end{center}
```



This text is left-aligned.  $\LaTeX$  is not trying to make each line the same length.

This text is right-aligned.  $\LaTeX$  is not trying to make each line the same length.

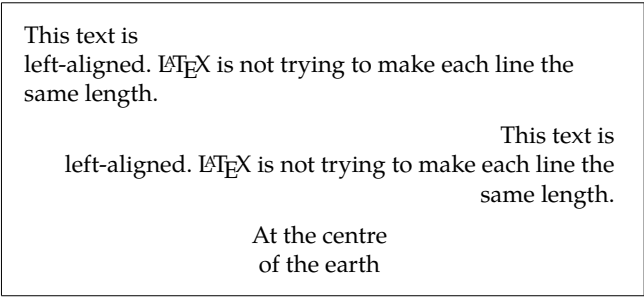
At the centre of the earth

위의 문단 정렬 환경들은 환경이 시작하기 바로 전과 환경이 끝난 바로 후에 약간의 빈 줄을 넣는다. 이러한 빈 공간을 원하지 않는다면 환경 대신 다음 예와 같이 라텍 명령을 사용하면 된다. `{\raggedright... \par}` 명령은 `flushleft` 환경과 마찬가지로 왼쪽 정렬을 주고, `{\raggedleft... \par}` 명령은 `flushright` 환경과 같이 오른쪽 정렬을 준다. 가운데 정렬에 해당하는 명령은 `{\centering... \par}`이다. 세 가지 명령 모두 `\par`로 끝난다는 점에 유의하라.

```
{\raggedright
This text is\\ left-aligned.
\LaTeX{} is not trying to make
each line the same length.\par}

{\raggedleft
This text is\\ left-aligned.
\LaTeX{} is not trying to make
each line the same length.\par}

{\centering
At the centre\\of the earth\par}
```



This text is left-aligned.  $\LaTeX$  is not trying to make each line the same length.

This text is left-aligned.  $\LaTeX$  is not trying to make each line the same length.

At the centre of the earth

**4.2 인용구** 인용구나 중요한 구절 및 예문 등은 `quote` 또는 `quotation` 환경을 사용한다. 두 환경의 차이는 문단 들여쓰기 (indent)에 있다. 들여쓰기를 하지 않는 `quote` 환경에 반해 들여쓰기를 하는 `quotation` 환경은 여러 문단에 걸친 긴 인용문을 처리할 때에 주로 쓴다. 두 환경 모두 환경의 시작 직전과 환경이 끝난 직후에 어느 정도의 빈 줄을 자동으로 넣는다.

```
A typographical rule of thumb for
the line length is:
\begin{quote}
On average, no line should be
longer than 66 characters.
\end{quote}
This is why \LaTeX{} pages have such large
borders by default and also why multicolumn
print is used in newspapers.
```

A typographical rule of thumb for the line length is:

On average, no line should be longer than 66 characters.

This is why  $\LaTeX$  pages have such large borders by default and also why multicolumn print is used in newspapers.

인용구를 넣을 때 마지막에 출처를 다는 것이 일반적이다. memoir 클래스를 사용하는 경우 다음 예와 같이 `\sourceatright{...}` 명령으로 출처를 쉽게 넣을 수 있다.

```
\begin{quotation}
This quotation has a short last line so there
is enough space for the source to be set at the
end of the line.\sourceatright{I. M. Short}
\end{quotation}

\begin{quotation}
The last line in this case turns out to be
too long for the source to be set at the end,
so it is set flushright on the following
line.\sourceatright{N. O. Space}
\end{quotation}
```

This quotation has a short last line so there is enough space for the source to be set at the end of the line. I. M. Short

The last line in this case turns out to be too long for the source to be set at the end, so it is set flushright on the following line. N. O. Space

**4.3 시** 시를 조판하는데 사용하는 라텍 환경은 `verse`이다. 각 행은 강제 줄바꿈 명령(`\`)으로, 그리고 각 연은 새로운 문단을 시작하는 빈 줄로 구분한다. 다음 예는 memoir 클래스에서 제공하는 확장된 형태의 `verse` 환경을 보여준다. `verse` 환경에 옵션인자로 주어진 길이는 시의 폭에 해당하는 값이다.

```
\PoemTitle*{A Limerick}
\begin{verse}[5cm]
There was a young man of Quebec \
Who was frozen in snow to his neck. \
\vin When asked: `Are you friz?' \
\vin He replied: `Yes, I is, \
But we don't call this cold in Quebec.'
\end{verse}
```

A Limerick

There was a young man of Quebec  
Who was frozen in snow to his neck.  
When asked: 'Are you friz?'  
He replied: 'Yes, I is,  
But we don't call this cold in Quebec.'

시의 제목은 `\PoemTitle{...}` 명령으로 주는데, 제목 앞에 자동으로 매겨지는 번호를 표시하지 않으려면 대신 `\PoemTitle*{...}` 명령을 쓴다. 시에서 들여쓰기를 주고 싶다면 위와 같이 `\vin` 명령을 사용하면 된다.

```
\begin{verse}
누구에게 감사받을 생각도 없이 \
나는 나에게 황홀을 느낄 뿐이다 \
나는 하늘을 찌를 때까지 \
자라려고 한다 \
무성한 가지와 그늘을 퍼려고 한다
\end{verse}
{\raggedleft\footnotesize 김윤성 詩 『나무』에서
\par}
```

누구에게 감사받을 생각도 없이  
나는 나에게 황홀을 느낄 뿐이다  
나는 하늘을 찌를 때까지  
자라려고 한다  
무성한 가지와 그늘을 퍼려고 한다

김윤성 詩 『나무』에서

**4.4 일반적인 리스트 환경** 라텍은 기본으로 세 가지 리스트 환경을 제공한다. 가장 일반적인 형태의 나열 방식은 각 항목들 앞에 동일한 마크를 붙이는 것이다. 이러한 나열은 `itemize` 환경으로 만든다. 다음 예와 같이 `itemize` 환경은 중첩해서 네 단계까지 사용할 수 있다.

```
\begin{itemize}
\item The label for the first level is a black dot, a bullet.
  \begin{itemize}
    \item That of the second level is an en dash.
      \begin{itemize}
        \item That of the third level is an asterisk.
          \begin{itemize}
            \item And the label for the fourth level is a centered dot.
          \end{itemize}
        \end{itemize}
      \end{itemize}
    \end{itemize}
  \end{itemize}
```

- The label for the first level is a black dot, a bullet.
  - That of the second level is an en dash.
    - \* That of the third level is an asterisk.
      - And the label for the fourth level is a centered dot.

각 단계마다 미리 설정된 마크가 표시되는데 `itemize` 환경에 옵션인자를 주면 마크를 바꿀 수 있다.

```
\begin{itemize}[S]
\item The text in the entries is of any length.
\item Successive entries are separated from one another by additional vertical spacing.
\end{itemize}
```

- § The text in the entries is of any length.
- § Successive entries are separated from one another by additional vertical spacing.

`itemize` 환경의 각 항목들은 적당한 수직 간격으로 나누어진다. 이 수직 간격은 단계에 따라 달라진다. 만일 수직 간격이 전혀 없는 촘촘한 형태의 나열이 필요하다면 `itemize` 환경을 시작한 직후 `memoir` 클래스가 제공하는 `\tightlist` 명령을 주도록 하자. 기본 수직 간격보다 조금 줄어들길 원하면 대신 `\firmlist` 명령을 사용하면 된다. 다음 예에서 수직 간격의 변화를 확인할 수 있다.

```
\begin{itemize}\tightlist
\item The text in the entries is of any length.
\item Successive entries are separated from one another by additional vertical spacing.
\end{itemize}
\begin{itemize}\firmlist
\item The text in the entries is of any length.
\item Successive entries are separated from one another by additional vertical spacing.
\end{itemize}
```

- The text in the entries is of any length.
- Successive entries are separated from one another by additional vertical spacing.
- The text in the entries is of any length.
- Successive entries are separated from one another by additional vertical spacing.

**4.5 번호가 매겨진 리스트 환경** 마크 대신 번호가 자동으로 매겨지는 나열은 `enumerate` 환경으로 만든다.

```
\begin{enumerate}
\item The labels consist of sequential numbers.
\item The numbering starts at 1 with every call to the enumerate environment.
\end{enumerate}
```

1. The labels consist of sequential numbers.
2. The numbering starts at 1 with every call to the enumerate environment.

`enumerate` 환경도 `itemize` 환경과 같이 중첩해서 네 단계까지 사용할 수 있다. 번호가 표시되는 방식은 단계에 따라 다르다. 다음 예는 `memoir` 클래스에 설정된 기본 표시 방식을 보여준다.

```

\begin{enumerate}
\item The numbering at the first level is with
  Arabic numerals followed by a period.
  \begin{enumerate}
\item At the second level, it is with
  lowercase letters in parentheses.
  \begin{enumerate}
\item The third level is numbered with
  lowercase Roman numerals with a period.
  \begin{enumerate}
\item At the fourth level, capital
  letters are used.
\item The label style can be changed.
\end{enumerate}
\item Back to the third level.
\end{enumerate}
\item Back to the second level.
\end{enumerate}
\item And we are at the first level again.
\end{enumerate}

```

1. The numbering at the first level is with Arabic numerals followed by a period.
  - a) At the second level, it is with lowercase letters in parentheses.
    - i. The third level is numbered with lowercase Roman numerals with a period.
      - A. At the fourth level, capital letters are used.
      - B. The label style can be changed.
    - ii. Back to the third level.
  - b) Back to the second level.
2. And we are at the first level again.

memoir 클래스는 번호가 표시되는 방식을 쉽게 바꿀 수 있도록 `enumerate` 환경을 확장시켰다.

```

\begin{enumerate}[i]
\item The first entry. \item The second entry.
\end{enumerate}
\begin{enumerate}[(a)]
\item The first entry. \item The second entry.
\end{enumerate}
\begin{enumerate}[\bfseries 1.]
\item The first entry. \item The second entry.
\end{enumerate}
\begin{enumerate}[A$^*$$]
\item The first entry. \item The second entry.
\end{enumerate}

```

- i) The first entry.
  - ii) The second entry.
- (a) The first entry.
  - (b) The second entry.
1. The first entry.
  2. The second entry.
- A\* The first entry.  
B\* The second entry.

`itemize` 환경과 마찬가지로 memoir 클래스가 제공하는 `\tightlist` 명령과 `\firmlist` 명령을 사용해 항목들 사이의 수직 간격을 줄일 수 있다.

```

\begin{enumerate}\tightlist
\item The labels consist of sequential numbers.
\item The numbering starts at 1 with every call
  to the enumerate environment.
\end{enumerate}

```

1. The labels consist of sequential numbers.
2. The numbering starts at 1 with every call to the enumerate environment.

#### 4.6 서술을 위한 리스트 환경

라벨과 이에 대한 서술이 나열되는 방식에 적합한 리스트 환경은 `description` 이다. 이 환경에서 라벨은 `\item` 명령의 옵션인자로 들어간다.

```

\begin{description}
\item[Stupid] things will not become smart
  because they are in a list.
\item[Smart] things, though, can be presented
  beautifully in a list.
\end{description}

```

- Stupid** things will not become smart because they are in a list.
- Smart** things, though, can be presented beautifully in a list.