

# The Not So Short Introduction to $\text{\LaTeX} 2_{\epsilon}$

Chapter 1~2

알아두어야 할 기본 사항  
텍스트의 조판

KINS

2013년 1월 9일

# Overview

*TEX이 너희를 자유케 하리라.*

# Overview

*TEX이 너희를 자유케 하리라.*

## 목표

L<sup>A</sup>T<sub>E</sub>X을 자유자재로 사용하는 방법과, L<sup>A</sup>T<sub>E</sub>X을 이용해서 큰 규모의 문서 작성을 하는 방법을 배운다.

# Overview

## 교과서?

- ▶ lshort-kr. (주교재)  
L<sup>A</sup>T<sub>E</sub>X을 처음 접하는 초보자부터 중급자까지 모두가 볼 수 있는 대단히 좋은 참고서.
- ▶ 한글과 T<sub>E</sub>X by 고기형, L<sup>A</sup>T<sub>E</sub>X Companion by Goossens, et al.  
L<sup>A</sup>T<sub>E</sub>X중급 사용자가 고급 사용자로 변신하기 위해 꼭 봐야 하는 책.
- ▶ The T<sub>E</sub>Xbook by Knuth.  
T<sub>E</sub>X의 바이블. 고급 사용자에게 필독서.

# Overview

- ▶ 기본 사항, 텍스트의 조판
  - ▶  $\text{T}\text{E}\text{X}$ 의 구조, 명령행작업, 문서클래스, 환경, 텍스트조판, 간격문제, 상호참조, 텍스트글꼴, 기본 명령어
- ▶ 수식의 조판, 특별한 기능
  - ▶ 수식 조판, 행렬조판, 조건식 조판, 괄호의 사용, 정리와 법칙, 수학글꼴문제,  $\text{BIB}\text{T}\text{E}\text{X}$ , 하이퍼링크
- ▶ 그림 그리기
  - ▶  $\text{picture}$ 환경,  $\text{ipe}$ 의 사용
  - ▶  $\text{X}\text{y-pic}$ 의 사용
- ▶ 고급기능
  - ▶ 여백조절
  - ▶ 새로운 환경, 새로운 명령어,
  - ▶ 박스
  - ▶ 카운터의 사용
- ▶ Project

## 제 1장

### 알아두어야 할 기본 사항

어떻게 읽어야 하는가?

TEX

# 어떻게 읽어야 하는가?

TEX

- ▶ 텍스 (X)



# 어떻게 읽어야 하는가?

$\text{T}_{\text{E}}\text{X}$

- ▶ 텍스 (X)
- ▶ 텍 (O)

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

# 어떻게 읽어야 하는가?

$\text{T}_{\text{E}}\text{X}$

- ▶ 텍스 (X)
- ▶ 텍 (O)

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

- ▶ 라텍스 (X)

# 어떻게 읽어야 하는가?

## T<sub>E</sub>X

- ▶ 텍스 (X)
- ▶ 텍 (O)

## L<sup>A</sup>T<sub>E</sub>X

- ▶ 라텍스 (X)
- ▶ 레이텍,라텍 (O)

# T<sub>E</sub>X? L<sup>A</sup>T<sub>E</sub>X?

- ▶ T<sub>E</sub>X: 도널드 커누스가 만든 조판 엔진, 몹시 원시적이어서 쓰기 어려움.
- ▶ L<sup>A</sup>T<sub>E</sub>X: 일반인도 편리하게(!) T<sub>E</sub>X을 쓸 수 있도록 만든 명령어 패키지.
- ▶ X<sub>E</sub>T<sub>E</sub>X, MacT<sub>E</sub>X, ChemT<sub>E</sub>X...





## 왜 $\LaTeX$ 을 쓰는가?

따라서  $\LaTeX$ 을 쓰면 다음과 같은 장점이 있다.

- ▶ 글의 내용과 논리적 구조에만 집중할 수 있다. 내용을 쓰고  $\LaTeX$ 에게 논리적 구조만 선언해주면 나머지는  $\LaTeX$ 이 알아서 해주니까!
- ▶ 이렇게 얻는 결과물은 아주 전문적이고 깔끔하다.
- ▶  $\LaTeX$ 은 사용자가 논리적으로 글을 쓸 수밖에 없도록 유도한다.
- ▶  $\TeX$ 은 아주 안정적이고 거의 모든 시스템에서 같은 방식으로 작동하므로, 이식성이 좋다.
- ▶ 차례, 색인, 상호참조, 번호붙이기, 참고문헌 등의 복잡한 문서 구조를 아주 쉽게 만들 수 있다.
- ▶ 수식이 아름답다.  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\LaTeX$ 을 사용하면 못 쓰는 수식이 없다.
- ▶ 뭔가있어보인다.

# 왜 $\LaTeX$ 을 쓰는가?

물론 단점도 있다.

- ▶ 어렵다. 디버깅을 해야 한다.
- ▶ 결과물을 바로 볼 수 없어서 답답하다.
- ▶ 주어진 서식에 해당하지 않는 문서는 만들기 어렵다. 즉, 결과물의 모양이 거의 똑같다.
- ▶ 주어진 서식에 손을 대려고 하면 영 좋지 않은 일이 종종 발생한다.
- ▶ 그림을 그리는 데 있어서 상당한 인내심을 요한다.
- ▶ Beamer와 같이  $\LaTeX$ 을 쓰는 것이 오히려 비효율적인 경우가 있다.





- ▶ 연습문제 0. T<sub>E</sub>X시스템을 컴퓨터에 설치하시오.  
[Hint. <http://www.ktug.or.kr>]

# LaTeX 입력 파일의 형식

LaTeX의 거의 모든 문서는 다음과 같은 구조를 가진다.

```
\documentclass[  ]{  }
```

```
\usepackage{  }
```

```
...전처리부
```

```
\begin{document}
```

```
....본문
```

```
\end{document}
```

# LaTeX 입력 파일의 형식

LaTeX의 거의 모든 문서는 다음과 같은 구조를 가진다.

<code>\documentclass[ ]{ }</code>	<code>\documentclass[a4paper, 10pt]{report}</code>
<code>\usepackage{ }</code>	<code>\usepackage{amssymb}</code>
... 전처리부	<code>\usepackage{amsmath}</code>
	<code>\begin{document}</code>
<code>\begin{document}</code>	Hello, World!
	<code>\[</code>
.... 본문	$\int x^2 dx = \frac{1}{3}x^3 + C$
	<code>\]</code>
<code>\end{document}</code>	<code>\end{document}</code>

# L<sup>A</sup>T<sub>E</sub>X 입력 파일의 형식

- ▶ L<sup>A</sup>T<sub>E</sub>X은 .tex의 확장자를 가지는 plain ASCII text를 처리한다.
- ▶ .tex파일에서 모든 공백과 줄바꿈은 출력물에서 띄어쓰기의 형태로 나타난다. 다만 한 줄 이상의 빈 줄은 문단의 끝을 나타낸다.
- ▶ `\`, `#`, `,`, `^`, `~`, `-`, `$`, `%`, `&`, `{}` 는 T<sub>E</sub>X의 예약 문자이므로 이를 출력하려면 각각 `\backslash`, `\#`, `\^`, `\~`, `\_`, `\$`, `\%`, `\&`, `\{`, `\}`의 형태로 입력해야 한다.

# L<sup>A</sup>T<sub>E</sub>X 입력 파일의 형식

- ▶ L<sup>A</sup>T<sub>E</sub>X의 명령어는 대소문자를 구분하며 \뒤에 오는 한 단어의 문자 또는 하나의 특수문자의 형태이다.

`\emph{}`, `\chapter[]{}` , `\hspace{}`, `\frac{}{}`

`\int`, `\times`, `\sum`, `\alpha`

`\#`, `\\`, `\|`

이 때, {}는 반드시 입력해야 하는 변수이고, []는 필요에 의해 입력하는 선택사항이다.

- ▶ % 뒤에 오는 한 행은 주석처리되며 결과물에 나타나지 않는다.

# 문서 클래스

문서클래스는 `\documentclass[ option ]{ class }` 형태로 지정한다.  
주로 쓰이는 옵션과 클래스는 다음과 같다.

## option

- ▶ 10pt, 11pt, 12pt
- ▶ a4paper
- ▶ twocolumn
- ▶ twoside
- ▶ openright
- ▶ landscape
- ▶ titlepage

## class

- ▶ article
- ▶ report
- ▶ book
- ▶ beamer

# 문서 클래스

Option	설명
10pt, 11pt, 12pt	본문 글꼴 크기 지정. 기본값은 10pt.
a4paper	용지 크기 지정.
twocolumn	다단 편집. 기본값은 oneseide.
twoside	양면인쇄 여부 설정. article과 report는 oneseide가 기본값. book은 twoside가 기본값.
openright	새로운 part를 항상 홀수쪽에서 시작함. book에서는 기본값이 openright. report에서는 기본값이 openany.
landscape	용지방향을 가로로 함. 기본값은 portrait.
titlepage	article 에서 제목 페이지를 따로 만듦. 기본값은 notitlepage.



# 문서 클래스

class	설명
article	section 수준의 짧은 글을 쓰기 위한 클래스.
report	chapter 수준의 비교적 긴 글을 쓰기 위한 클래스.
book	part수준의 아주 긴, 책을 만들기 위한 클래스.
beamer	여러분이 지금 보고 있는 자료들을 만들기 위한 클래스.

# 패키지

- ▶ 패키지는  $\LaTeX$ 의 성능을 향상시키기 위한 명령어 묶음이다.
- ▶ 입력파일의 전처리부에 `\usepackage[option]{package}` 형태로 로드할 수 있다.

## 자주 사용되는 패키지

`\usepackage[hangul]{dhucs}`%TeX이 한글을 식자할 수 있게 해줌

`\usepackage{dhucs-ucshyper}`%한글 책갈피 생성하게 해 줌

`\usepackage{amsmath}`%각종 수학과 관련된 환경 제공

`\usepackage{amssymb}`%각종 수학기호와 수학기호를 사용

# 환경

호출이름이 name인 환경은

```
\begin[option]{name}{ factor1 }{ factor 2 } ...{ factor n }  
...  
  
\end{name}
```

형태로 호출하며 \begin과 \end사이에 있는 내용에 특정한 서식을 준다. 이 때, [] 안에 들어 있는 option은 필요에 의해 입력하는 선택사항이지만, {} 안에 있는 factor는 그 환경의 인자 혹은 변수라고 불리며 **반드시 입력해야** 하는 것들이다.

## 제 2장

### 텍스트의 조판

# 기본 사항

기본적으로 텍스트의 내용은 입력 파일에서 작성한 그대로 나타난다.  $\LaTeX$ 이 조정해주는 것은, 여백, 자간, 행간등의 **글자 배치**뿐이다.

- ▶  $\LaTeX$ 은 주어진 문서클래스와 레이아웃에 의해서 사용자가 입력한 텍스트를 보기좋은 모양으로 배치해준다.
- ▶ 별다른 설정이 없으면 모든 문단은 양쪽정렬된다.
- ▶ 별다른 지시가 없으면 새로운 문단의 시작은 자동으로 들여쓰기된다.
- ▶ 입력파일의 모든 여백은 하나의 띄어쓰기로 나타난다. 다만, 한 줄 이상의 빈 행은 문단의 끝을 나타낸다.

# 기본 사항-예시

안녕하세요. 정흥택입니다.  
\LaTeX은 정말로  
재미있군요. 앞으로  
열심히 공부  
하겠습니다.

안녕하세요. 박승쿵입니  
다.  
오늘은 \today 이군요.

안녕하세요. 정흥택입니다.  
L<sup>A</sup>T<sub>E</sub>X은 정말로 재미있군요.  
앞으로 열심히 공부하겠습  
니다.

안녕하세요. 박승쿵입니다.  
오늘은 2013년 1월 9일이군  
요.

## 줄바꿈과 쪽나눔

강제로 개행하거나 새 쪽을 시작하고 싶을 때 사용하는 명령은 다음과 같다.

명령어	설명
<code>\\[length]</code>	개행하며 새로운 행 위로 length만큼의 여백을 준다.
<code>\\*[length]</code>	<code>\\</code> 와 같지만 새 행이 이전 행과 같은 쪽에 나오도록 한다.
<code>\newpage</code>	새로운 페이지를 시작한다. 또는 현재 단을 끝내고 새로운 단을 시작한다.
<code>\par</code>	하나의 빈 행과 같은 기능. 문단을 끝내고 새로운 문단을 시작한다.

# 줄바꿈과 쪽나눔-예시

안녕하세요. 정흥택입니  
다.  $\LaTeX$ 은 정말로  
재미있군요. 앞으로 열  
심히 공부하겠습니다.  
다.  $\ll[-1em]$  안녕하  
세요. 박승쿵입니다.

안녕하세요. 정흥택입니다.  
 $\LaTeX$ 은 정말로 재미있군요.  
앞으로 열심히 공부하겠습  
니다.  
안녕하세요. 박승쿵입니다.



## 계층 구조

모든 장문의 글은 논리적 계층구조를 가지는데, article 클래스의 경우 문서는

```
\section{...}-\subsection{...}-\subsubsection{...}
```

으로 구성되며, book 과 report 클래스의 경우

```
\chapter{...}-\section{...}
                    -\subsection{...}-\subsubsection{...}
```

로 구성된다. 각 계층의 번호는 자동으로 부여된다.

{ }안에는 각 계층의 title을 적을 수 있다. 또한, book클래스의 경우 계층 번호에 영향을 주지 않는 \part{...}라는 명령이 있다.

## 제목과 목차 만들기

다음 명령어들은 제목을 생성하기 위해 필요한 정보들을  $\LaTeX$ 에게 제공한다.

명령어	설명
<code>\title</code>	문서의 제목을 지정. <code>\thanks{}</code> 명령어를 써서 주석을 달 수 있음.
<code>\author</code>	저자 지정. 여러명인 경우 <code>\and</code> 명령어를 써서 구분.
<code>\date</code>	날짜 지정. 지정하지 않으면 컴파일 한 날짜를 찍어줌.

위와 같이 정보를 입력한 후, `\maketitle`명령어를 쓰면 명령어가 쓰인 자리에 제목을 생성함.

`\tableofcontents`명령어를 사용하면 명령이 사용된 자리에 목차를 생성함.

# 제목과 목차 만들기-예시

```
\title{\LaTeX입  
문\thanks{KAIST의 지  
원을 받아 작성}}  
\author{정흥  
택\footnote{수  
학과} \and 정흥  
택\footnote{물리과}}
```

```
\maketitle
```

## LaTeX입문<sup>1</sup>

정흥택<sup>a</sup> 정흥택<sup>b</sup>

KINS

2013년 1월 9일

---

<sup>a</sup>수학과

<sup>b</sup>물리과

<sup>a</sup>KAIST의 지원을 받아 작성

## 각주와 여백글 만들기

- ▶ 각주는 `\footnote{...}` 명령을 통해 만들 수 있다.
- ▶ 문서의 여백에 글을 쓰고자 할 땐,  
`\marginpar[even content]{odd content}` 명령어를 쓰면 된다.  
이 때, even(odd) content는 각각 해당 페이지가 짝수(홀수) 페이지인 경우 여백에 나타낼 내용이다.

## 쪽 양식 지정

페이지 양식은 `pagestyle{ style }` 명령을 **전처리부**에 써서 지정한다. `style`에 올 수 있는 인자는 다음과 같다.

style	설명
plain	문서 아래 중앙에 쪽 번호 식자.
empty	아무것도 식자하지 않음
headings	머리글에 쪽 번호와 장,절 제목을 식자.

특정 페이지의 쪽 양식만 바꾸기 위해서는 원하는 위치에 `\thispagestyle{ style }` 명령을 쓴다.

## 기타 명령어

명령어	설명
<code>\emph{...}</code>	{ }안의 객체를 <b>강조emph</b> 하여 식자해준다.
<code>\underline{...}</code>	{ }안의 객체에 <u>밑줄을</u> 그어준다.
<code>'...', '‘...’'</code>	'작은 따옴표'와 "큰 따옴표" 를 찍어준다.
<code>\"{..}</code>	Schrödinger 와 같이 { }안의 문자에 악센트 기호를 단다.
<code>\noindent</code>	문단의 시작위치에 쓰면 문단 들여쓰기를 하지 않는다.
<code>~</code>	단어 사이의 여백에 쓰면 그 여백은 줄바꿈에 의해 잘리지 않는다.
<code>\@</code>	문장이 끝났음을 나타낸다.

# 예시

```
\title{이것은 연습장}
```

```
\author{잉여킹}
```

```
\maketitle
```

```
\tableofcontents
```

```
\chapter{우왕}
```

이것은  $\LaTeX$ 의 힘이다!  
\footnote{엄청 나죠?}

```
\section{Good}
```

```
\underline{이야}
```

```
\subsection{엄청 나군!}
```

“종다!”

```
\subsection{Schrödinger 방정식}
```

은 어렵습니다.  
\footnote{진짜 어려워요 \ldots} **\emph{양자역학}**~시간에 배우죠.

```
\chapter{킹왕짱}
```

역시나  $\LaTeX$ 이 최고지!!

## Remark

- ▶ 상호참조를 사용하는 명령(`\tableofcontents`, `\href`, `\footnote` 등)은 여러번의 컴파일을 해야 번호가 정상적으로 표시된다.
- ▶ 각 장절 명령에는 대응하는 '별표명령'이 있다. 별표명령은 일반 명령과 똑같지만 번호 붙이기를 하지 않는다.

```
\chapter*{...}, \section*{...}
```

```
\subsection*{...}, \subsubsection*{...}
```

- ▶ book클래스의 경우, `\frontmatter`, `\mainmatter`, `\appendix`, `\backmatter` 등의 명령어를 제공한다. 직접 효과를 확인해보기 바란다.



- ▶ 연습문제 1. 별표 붙은 명령의 효과와 `\frontmatter`, `\mainmatter`, `\appendix`, `\backmatter`의 효과를 확인하라. 책의 머릿말(Preface) 부분은 어떻게 만들면 될까?
- ▶ 연습문제 2. `book` 클래스에서 옵션인자로 `oneside`를 줄 때와 `twoside` 줄 때의 차이를 확인하라.

## 환경-List

항목 나열과 관련된 환경은 세 가지가 있다.

<code>\begin{itemize}</code>	<code>\begin{enumerate}</code>	<code>\begin{description}</code>
<code>\item[...]</code> ....	<code>\item</code> ....	<code>\item[...]</code> ....
<code>\item[...]</code> ....	<code>\item</code> ....	<code>\item[...]</code> ....
...	...	...
<code>\end{itemize}</code>	<code>\end{enumerate}</code>	<code>\end{description}</code>

: 단순 항목 나열

: 번호 붙은 나열

: 사전식 항목 설명

이 때, [...]안에는 항목 나열에 쓰일 기호 혹은 항목 설명의 표제어가 들어갈 수 있다.

# 환경-List

```
\begin{itemize}
\item 이렇게
\item[@] 사용
\item 합니다
\begin{itemize}
\item 중첩도
\item 가능 하고요
\end{itemize}
\item 계속 이어서 나옵니다.
\end{itemize}
```

- ▶ 이렇게
- ◎ 사용
- ▶ 합니다
  - ▶ 중첩도
  - ▶ 가능하고요
- ▶ 계속 이어서 나옵니다.

# 환경-List

```
\begin{enumerate}
\item 이렇게
\item 사용
\item 합니다
\begin{enumerate}
\item 중첩도
\item 가능하고요
\end{enumerate}
\item 계속 이어서 나옵니다.
\end{enumerate}
```

1. 이렇게
2. 사용
3. 합니다
  - 3.1 중첩도
  - 3.2 가능하고요
4. 계속 이어서 나옵니다.

## 환경-인용

인용에 쓰이는 환경은 두 가지가 있다.

```
\begin{quote}
```

짧은 인용문

```
\end{quote}
```

```
\begin{quotation}
```

기이이이이이인 인용문

....

```
\end{quotation}
```

quotation환경은 단락 들여쓰기를 해 주지만 quote환경은 그렇지 않다.

## 환경-인용

히로나카 헤이스케 형님은 이렇게 말씀하셨다:

```
\begin{quote}
```

난 바보이다.

```
\end{quote}
```

정말 옳은 말이다.

히로나카 헤이스케 형님은 이렇게 말씀하셨다:  
*난 바보이다.*

정말 옳은 말이다.

## 환경-tabular

tabular는 표를 만드는 환경이다.

```
\begin{tabular}[ pos ]{ format }  
contents  
\end{tabular}
```

```
\begin{tabular*}{ width }[ pos ]{ format }  
contents  
\end{tabular*}
```

## 환경-tabular

```
\begin{tabular}[ pos ]{ format } \end{tabular}
```

```
\begin{tabular*}{ width }[ pos ]{ format } \end{tabular*}
```

- ▶ pos인자는 표의 baseline을 기준으로 한 수직 위치를 결정한다.

pos 인자	설명
t	현재 행의 baseline과 표의 윗선을 맞춤.
b	현재 행의 baseline과 표의 아래선을 맞춤.
c	현재 행의 baseline과 표의 중심을 맞춤.

- ▶ width인자는 표의 전체 폭을 결정한다.



## 환경-tabular

```
\begin{tabular}[ pos ]{ format } \end{tabular}
```

```
\begin{tabular*}{ width }[ pos ]{ format } \end{tabular*}
```

- ▶ format인자는 표가 가지는 열의 수와 수직선의 모양을 결정한다.

format 인자	설명
c	해당 열의 내용을 가운데 정렬함.
l	해당 열의 내용을 왼쪽 정렬함.
r	해당 열의 내용을 오른쪽 정렬함.
p{ width }	해당 열의 폭을 width로 고정하고 내용은 자동 정렬함. 필요한 경우 줄바꿈을 할 수 있음.
	해당 위치에 열을 나누는 수직선을 그음.
	해당 위치에 열을 나누는 이중 수직선을 그음.
@{ }	{ }안의 문자를 열이 바뀔 때 마다 표에 식자함.

## 환경-tabular

표의 내용은 다음과 같이 채운다.

명령어	설명
<code>\\</code>	해당 행을 종료함. 행의 끝을 표시.
<code>&amp;</code>	열을 구분함. 각 행에는 반드시 (format인자에서 정의한 열의 수-1)개의 &가 나와야 한다.
<code>\hline</code>	행 구분선을 그음.
<code>\cline{i-j}</code>	i번째 열과 j번째 열 사이에 수평 행 구분선을 그음.
<code>\vline</code>	해당 위치에 행의 높이를 길이로 하는 수직선을 그음.

## 환경-tabular

복잡한 표를 만들기 위해서는 여러 열을 **하나의 열**로 묶을 필요가 있는데, 이러한 일은

```
\multicolumn{ n }{ format }{ contents }
```

명령으로 할 수 있다. 이 명령은 &바로 뒤에만 쓰여야 한다.

- ▶ n인자는 1 이상의 자연수이고, 묶을 열의 수를 지정.
- ▶ format인자는 묶인 열의 형식을 지정. tabular의 format인자와 같음.
- ▶ contents인자는 묶인 열에 쓸 내용을 입력.

## 환경-tabular

```
\begin{tabular}{r|cc}  
\hline \hline  
& 키 & 나이 \\ \hline  
정 흥 택 & 180cm & 20 \\ \hline  
박 승 쿵 & 175cm & 19 \\ \hline  
\cline{3-3}  
평균 & & 19.5 \\ \hline \hline  
\end{tabular}
```

	키	나이
정흥택	180cm	20
박승쿵	175cm	19
평균		19.5

## 환경-tabular

```
\begin{tabular}{c|c|c|p{3em}}  
\hline  
& \multicolumn{2}{|c|}{GPA} & \\  
\cline{2-3}  
& 종합 & 전공 & 비고 \\  
\hline  
정흥택 & 3.2 & 3.3 & 장짤\\  
\hline  
장선생 & 4.3 & 4.3 & 과탐, 장  
학금 \\  
\hline  
\end{tabular}
```

	GPA		비고
	종합	전공	
정흥택	3.2	3.3	장짤
장선생	4.3	4.3	과탐, 장학금



- ▶ 연습문제 5. 소수점 정렬에 유의해서 다음 표를 식자하라.

MAS101 최종점수	
학번	점수
20080344	199.81
20090001	95.0
20090886	100.99
20130001	3.14
20139999	183.90
만점	200.0

[HINT. @{} 를 응용하라.]

# 환경-Abstract

abstract 환경은 말 그대로 '요약'을 위한 환경으로 주로 article 클래스에서 사용된다.

```
\begin{abstract}
```

더 이상의 자세한 설명은 생략한다.

```
\end{abstract}
```

요약

더 이상의 자세한 설명은 생략한다.



## 환경-정렬

정렬환경엔 `flushright`, `flushleft`, `center` 가 있으며, 각각은 텍스트를 우측정렬, 좌측정렬, 중앙정렬해준다.

## 떠다니는 개체(Floating Object)

문서에서 표와 그림을 효과적으로 배치하기 위한 방법.

<code>\begin{figure}[ spec ]</code>	<code>\begin{table}[ spec ]</code>
그림	<code>\begin{tabular}{...}</code>
<code>\caption{....}</code>	...
<code>\end{figure}</code>	<code>\end{tabular}</code>
	<code>\caption{....}</code>
	<code>\end{table}</code>

## 떠다니는 개체(Floating Object)

$\LaTeX$ 은 떠다니는 개체를 대기열에 둔 후 **먼저 선언된 것 부터 차례대로 출력**한다. 이 때, 떠다니는 개체가 본문 안에서 적당한 위치를 차지하도록 배치되는데, `spec`인자로 그 위치를 어느정도 조절할 수 있다.

### 위치 지정자의 종류와 의미

- ▶ `!`: 출력 모양이 좋지 않아도 상관 없다.
- ▶ `h`: 선언된 바로 그 위치에 출력해라.
- ▶ `t`: 선언된 페이지의 맨 위에 출력해라.
- ▶ `b`: 선언된 페이지의 맨 아래에 출력해라.
- ▶ `p`: 떠다니는 개체를 위한 특별한 쪽을 만들어 거기에 배치해라.

위치 지정자가 명시되지 않으면  $\LaTeX$ 은 기본값 `[tbp]`를 사용한다.

## 떠다니는 개체(Floating Object)

예컨대, 사용자가 떠다니는 개체를

```
\begin{table}[!htb]  
...  
\end{table}
```

로 선언했다면, L<sup>A</sup>T<sub>E</sub>X은 이 개체를 보기 좋던 안 좋던, 선언된 그 위치에 배치하려 할 것이며, 여의치 않으면 그 페이지의 맨 위 그도 아니면 그 페이지의 맨 아래에 배치한다.

이 모든 경우가 다 불가능하면 개체는 출력이 안 되고, 따라서 그 뒤의 떠다니는 개체 역시 모두 출력되지 않는다.

## 떠다니는 개체(Floating Object)

caption과 그 번호는 L<sup>A</sup>T<sub>E</sub>X이 표와 그림을 구분하여 알아서 붙여준다.

```
\begin{table}[!ht]
\begin{tabular}{r|cc}
\hline \hline
& 키 & 나이 \\
\hline
정흥택 & 180cm & 20 \\
박승쿵 & 175cm & 19 \\
\hline \hline
\end{tabular}
\caption{이것은 캡션}
\end{table}
```

	키	나이
정흥택	180cm	20
박승쿵	175cm	19

표: 이것은 캡션

## 떠다니는 개체(Floating Object)

다단 편집 시 양단에 걸치는 큰 표나 그림을 떠다니는 개체로 만들기 위해서는

```
\begin{figure*}[ spec ] \end{figure*}
\begin{table*}[ spec ] \end{table*}
```

명령을 사용할 수 있다.

## 상호 참조

어떤 것을 참조하고 싶을 때, `\label{ marker }` 명령과 `\ref{ marker }`를 사용한다.

- ▶ `\label{marker}`를 사용하여 나중에 참조할 객체에 특별한 `marker`를 부여한다.
- ▶ marking한 부분을 참조하려면 해당하는 부분의 `marker`에 대해 `\ref{marker}`명령을 쓴다.
- ▶ 그러면  $\text{\LaTeX}$ 은 `\ref{}`가 쓰인 위치에 marking된 부분의 장 번호, 절 번호, 그림 또는 표의 번호를 식자해준다.
- ▶ `\pageref{ marker }`명령어는 `marker`가 놓인 페이지를 식자해준다.

## 상호 참조-예시

```
\section{예제입니다.}
```

네 그렇습니다.

```
\section{이 절은 곧 인용될 것이야.}\label{go}
```

과연 그럴까요.

```
\section{아인슈타인 방정식}
```

앞의 `\ref{go}`절에서 이미 다른 내용입니다.



## Remark

- ▶ marker는 나중에 기억하기 쉬운 것으로 지정하는 게 좋다.
- ▶ `label{...}` 명령은 숫자를 붙여주는 명령어(`\chapter{}`, `\section{}`, `\caption{}` 등) 바로 뒤에 붙여야 잘 작동한다.
- ▶ 참조된 번호가 잘 식자되려면 최소 두 번의 컴파일이 필요하다.

# 글꼴 문제

L<sup>A</sup>T<sub>E</sub>X은 몇 가지 글꼴을 제공하고 있으며 크기를 조절할 수 있다.

## 크기 조절 명령어

- ▶ `\tiny`
- ▶ `\footnotesize`
- ▶ `\small`
- ▶ `\normalsize`
- ▶ `\large`
  
- ▶ `\LARGE`
  
- ▶ `\Huge`

## 글꼴 변환 명령어

- ▶ `\texttt{}:` typewriter
- ▶ `\textrm{}:` roman
- ▶ `\textsl{}:` *slanted*
- ▶ `\textbf{}:` **bold face**
- ▶ `\textsf{}:` san serif
- ▶ `\textsc{}:` SMALL CAPITAL
- ▶ `\textit{}:` *italic*
- ▶ `\textnormal{}:` document font

## 글꼴 문제-예시

```
{\tiny \texttt{글꼴  
을}} {\LARGE \textbf{\textsf{너  
무}}}  
{\small \textsl{자  
주}} {\normalsize 바꾸  
면} {\Large \textit{가독성  
이}} {\textup \footnotesize{떨  
어집니다.}}
```

글꼴을 **너무** 자주 바꾸면 **가독성이** 떨어집니다.

## Remark

- ▶ 글자 크기를 조절하는 명령어는 중괄호 {}내에서만 효과를 나타낸다. 중괄호 밖에서 쓰이면 문서 전체에 영향을 주므로 주의한다.
- ▶ 글꼴 변환 명령은 중첩해서 사용할 수 있다.
- ▶ 너무 다양한 글꼴을 사용하는 것은 좋은 생각이 아니다. 또한, 문서 안에서 글꼴을 직접 바꾸는 것 역시 좋은 생각이 아니다.
- ▶ 수학 글꼴을 바꾸는 것은 다음시간에 한다.
- ▶ 더욱 자세한 설명은 lshort 112, 113페이지를 참고.

- ▶ 숙제 1. 다음 문서를 만드시오.