

L^AT_EX 시작하기

기본적인 글쓰기

권현우

서강대학교



Figure: Donald E. Knuth (1938 – 현재)

1. Donald Knuth가 자신의 저서 “The Art of Computer Programming”을 출판하려고 만든 프로그램



Figure: Donald E. Knuth (1938 – 현재)

1. Donald Knuth가 자신의 저서 “The Art of Computer Programming”을 출판하려고 만든 프로그램
2. T_EX은 DTP(DeskTop Processing) 프로그램이다. 조판 프로그램이다. 처음 개발할 당시 목적도 책을 만들기 위함이었다.



Figure: Donald E. Knuth (1938 – 현재)

1. Donald Knuth가 자신의 저서 “The Art of Computer Programming”을 출판하려고 만든 프로그램
2. T_EX은 DTP(DeskTop Processing) 프로그램이다. 조판 프로그램이다. 처음 개발할 당시 목적도 책을 만들기 위함이었다.
3. 따라서 WYSIWYG기반의 워드프로세서랑 다르다.

- 마이클 스피박(Michael Spivak, 1983-1985) / AMS-TeX Macro
- 레슬리 램포트(Leslie Lamport, 1984) / L^ATeX Macro
- Hàn Thế Thành (90년대 초) / pdfTeX Engine
- 조나단 큐(Jonathan Kew, 2008) / XeTeX Engine
- 한스 하겐(Hans Hagen, 2007) / LuaTeX Engine
- 고기형(한TeX, 1990), 최우형-백윤주 (hlatex, 1992), 차재춘(hLaTeXp, 1995), 은광휘(hLaTeX, 1999), 김도현-김강수-은광휘(KoTeX, 2007), 조진환(xdvipdfmx, 2008)

- 배우는 그자리에서 한 문장이라도 제대로 만드는데 워드프로세서보다 시간이 많이 걸린다.

- 배우는 그자리에서 한 문장이라도 제대로 만드는데 워드프로세서보다 시간이 많이 걸린다.
- 제대로 `output`을 만들어도 어떻게 만들어낸 건지 이해가 안간다.

- 배우는 그자리에서 한 문장이라도 제대로 만드는데 워드프로세서보다 시간이 많이 걸린다.
- 제대로 `output`을 만들어도 어떻게 만들어낸 건지 이해가 안간다.
- 자신이 잘못 입력해서 에러가 났을 때 그 에러가 왜 일어난 건지 처음 단계에서는 잘 알 수 없다.

- 배우는 그자리에서 한 문장이라도 제대로 만드는데 워드프로세서보다 시간이 많이 걸린다.
- 제대로 `output`을 만들어도 어떻게 만들어낸 건지 이해가 안간다.
- 자신이 잘못 입력해서 에러가 났을 때 그 에러가 왜 일어난 건지 처음 단계에서는 잘 알 수 없다.
- 특별히 어떤 모양을 만들고 싶을 때 다른 사람의 설명을 들어도 워드처럼 직관적으로 와닿지 않다.

그럼에도 \TeX 이 쓸만한 이유

- 논리적인 문서작성

그럼에도 T_EX이 쓸만한 이유

- 논리적인 문서작성
- 아무리 오래되어도 T_EX파일 editing이 불가능하지 않다.

그럼에도 T_EX이 쓸만한 이유

- 논리적인 문서작성
- 아무리 오래되어도 T_EX파일 editing이 불가능하지 않다.
- 수식조판 할 때, 최고의 솔루션

그럼에도 T_EX이 쓸만한 이유

- 논리적인 문서작성
- 아무리 오래되어도 T_EX파일 editing이 불가능하지 않다.
- 수식조판 할 때, 최고의 솔루션
- 다국어조판

그럼에도 T_EX이 쓸만한 이유

- 논리적인 문서작성
- 아무리 오래되어도 T_EX파일 editing이 불가능하지 않다.
- 수식조판 할 때, 최고의 솔루션
- 다국어조판
- 프로그래밍

그럼에도 \TeX 이 쓸만한 이유

- 논리적인 문서작성
- 아무리 오래되어도 \TeX 파일 editing이 불가능하지 않다.
- 수식조판 할 때, 최고의 솔루션
- 다국어조판
- 프로그래밍
- 무료 조판프로그램

그럼에도 T_EX이 쓸만한 이유

- 논리적인 문서작성
- 아무리 오래되어도 T_EX파일 editing이 불가능하지 않다.
- 수식조판 할 때, 최고의 솔루션
- 다국어조판
- 프로그래밍
- 무료 조판프로그램
- 내용은 그대로, 형식은 자유자재로

그럼에도 $\text{T}_{\text{E}}\text{X}$ 이 쓸만한 이유

- 논리적인 문서작성
- 아무리 오래되어도 $\text{T}_{\text{E}}\text{X}$ 파일 editing이 불가능하지 않다.
- 수식조판 할 때, 최고의 솔루션
- 다국어조판
- 프로그래밍
- 무료 조판프로그램
- 내용은 그대로, 형식은 자유자재로
워드프로세서는 취미가 되기 힘들지만, $\text{T}_{\text{E}}\text{X}$ 은 취미가 될 수 있습니다.

- 수학 교과서
- 포스터
- 이력서
- 시험지
- 프로그램북
- 다국어 조판
- 발표자료

총 6강동안 진행할 이 강연 시리즈의 목표는 다음과 같다.

1. $\text{T}_\text{E}X$ 으로 한글이 담긴 과제 보고서를 만들 수 있다.
2. $\text{T}_\text{E}X$ 으로 학술논문을 작성할 수 있다.
3. $\text{T}_\text{E}X$ 으로 발표자료를 효과적으로 만들 수 있다.

- 이 강의에서 모든 걸 배울 수 있는 생각을 버리자.
- 있는 그대로 클래스에서 제공하는 규칙을 최대한 존중하자.
- 필요한 패키지를 만들려는 시도는 지금 하지 않는다.
- 주요한 패키지의 매뉴얼을 잘 읽자.
- 웬만한 질문의 답은 KTUG와 stackexchange에 있다.

금일 학습목표

1. \TeX 문서의 구조를 알고, 입력방법에 대해서 공부한다.
2. 간단한 문서를 작성한다.
3. 한글텍 사용을 해보고, 한글문서의 기본포맷인 Oblivoir를 공부한다.

TeX 문서 구조와 입력방법

```
\documentclass{article}
```

```
\usepackage{graphicx}
```

```
\begin{document}
```

```
Hello World!
```

```
\end{document}
```

- `\begin {document}` 이전까지를 **preamble**(전언)이라 부른다.
- `\begin {document}`와 `\end {document}`로 둘러싸인 부분은 실제 pdf파일에 출력될 부분이다.
- `\documentclass`서 부르는 것은 `article.cls` 파일이다. 이 명령어는 이 문서의 형식을 담은 명령어들이 모인 파일을 부르는 명령어다. 이런 파일을 클래스(class)라 부른다.
- `\usepackage`서 부르는 것은 그림파일을 삽입할 수 있게 하는 `graphicx.sty`이다. 패키지는 라텍의 문법으로 이루어진 명령어들의 모임으로 특정 목적을 위해 만들어진 파일이다.

TeX 문서 구조와 입력방법

```
\documentclass{article}
```

```
\usepackage{graphicx}
```

```
\begin{document}
```

```
Hello World!
```

```
\end{document}
```

클래스란?

- 문서의 형식을 결정한다.
- 예를 들어 article.cls은 chapter 명령어가 없는 일반 글이지만, book.cls은 chapter 명령어가 있는 '책'을 위한 클래스 파일이다.
- 대표적인 클래스 종류는 article, book, report, amsart, amsbook, memoir, oblivoir

TeX 문서 구조와 입력방법

```
\documentclass{article}
```

```
\usepackage{graphicx}
```

```
\begin{document}
```

```
Hello World!
```

```
\end{document}
```

스타일이란?

- \LaTeX 커널이 제공하는 기본 기능을 확장하거나 사용자의 명령을 모아놓은 일종의 명령 집합.
- 예를 들어, 그림을 넣는 것은 \LaTeX 에서 기본적으로 제공하는 것이 아니다! `graphicx` 패키지를 불러야 한다.
- 패키지가 너무 많아서 모든 것을 다 외우기 힘들. 주요 패키지를 익히고 그때그때 필요한 것을 찾아서 익히는 게 필요.

TeX 문서 구조와 입력방법

```
\documentclass{article}
```

```
\usepackage{graphicx}
```

```
\begin{document}
```

```
Hello World!
```

```
\end{document}
```

스타일이란?

- \LaTeX 커널이 제공하는 기본 기능을 확장하거나 사용자의 명령을 모아놓은 일종의 명령 집합.
- 예를 들어, 그림을 넣는 것은 \LaTeX 에서 기본적으로 제공하는 것이 아니다! `graphicx` 패키지를 불러야 한다.
- 패키지가 너무 많아서 모든 것을 다 외우기 힘들. 주요 패키지를 익히고 그때그때 필요한 것을 찾아서 익히는 게 필요.

요약하면

클래스는 문서의 바탕을 결정하고, 스타일은 여러가지 기능을 구현하는 걸 도와주는 장치

LaTeX의 명령어 종류

크게 command와 environment가 있다.

```
\command[option]{m1}{m2}
```

```
\begin{environment}[option]{m1}{m2}
```

....

```
\end{environment}
```

- option은 대괄호로 입력해야 한다. (명령어가 옵션을 지정한 경우)
- m을 입력하는 파트가 있다면 반드시 입력해야 한다. (명령어가 그리 지정한 경우)

LaTeX의 명령어 종류

크게 command와 environment가 있다.

```
\command[option]{m1}{m2}
```

```
\begin{environment}[option]{m1}{m2}
```

....

```
\end{environment}
```

- option은 대괄호로 입력해야 한다. (명령어가 옵션을 지정한 경우)
- m을 입력하는 파트가 있다면 반드시 입력해야 한다. (명령어가 그리 지정한 경우)

예제

다음 명령어들을 본문에 입력해보자.

```
\begin{center}
```

```
\LARGE Centering!   $\sqrt{2}$    $\sqrt[3]{2}$ 
```

```
\end{center}
```

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

* #은 L^AT_EX 프로그래밍과 연관이 있는 특수문자다. 이를 인자기호(parameter symbol)이라 부른다.

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

* \$은 수식 기호

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

* 은□ 위첨자, _은 아래첨자를 의미한다.

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

* \ 은 명령어 구분자다.

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

* {}은 그룹화 시키는 명령어다.

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

* 은□ 스페이싱과 연관이 있다.

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

* &은 표나 행렬에서 열을 구분해준다.

L^AT_EX의 명령어 종류

L^AT_EX을 사용하다보면 워드프로세서와 달리 몇 가지 규칙들이 있다.

- L^AT_EX은 공백문자를 연속으로 입력해도 한 개로 인식한다.
- 한 줄 공백을 주어야 문단 구분이 가능하다.
- 다음의 문자들을 그대로 입력할 경우 L^AT_EX은 에러를 일으킨다. 이는 라텍스에서 의미를 가지고 있기 때문이다.

\$ ^ _ & \ ~ { } %

* % 은 주석처리에 쓰는 명령어다.

따라서 앞에서 언급한 기호를 작성하고 싶다면, 다음과 같이 작성해야 한다.

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

그 외 참고할만한 특수문자들이 다양한데, 거의 모든 문자들은

```
> texdoc symbols-a4
```

따옴표

또 레이텍 문서를 작성할 때 주의해야 할 점이 따옴표인데, 워드프로세서에서 입력하는 습관대로 입력하면 안 된다.

‘작은 따옴표’ ‘‘큰 따옴표’’

한글텍 사용하기

```
\documentclass{article}
```

```
\usepackage{kotex}
```

```
\begin{document}
```

텍에서 한글 입력이 가능합니다.

```
\end{document}
```

한글텍 사용하기

```
\documentclass{oblivoir}
```

```
\begin{document}
```

한글문서에 최적화 된 패키지, 오블리무아(oblivoir)입니다.

obliviscence(망각, 잊기 쉬움)와 클래스 이름 memoir의 합성어입니다.

한글 문서를 작성할 때의 온갖 애로사항을 많이 해소해줍니다.

```
\end{document}
```

* 앞으로 이번 1강은 클래스 oblivoir를 기준으로 설명할 예정입니다.

기본적인 글쓰기

```
\documentclass{oblivoir}
```

```
\title{라텍 글쓰기}
```

```
\author{권현우}
```

```
\date{2022년}
```

```
\begin{document}
```

```
\maketitle
```

안녕하세요.

```
\end{document}
```

문서의 계층구조

```
\chapter[짧은 제목]{챕터 이름} % \documentclass[chapter]{oblivoir}
```

```
\section[짧은 제목]{절 이름}
```

```
\subsection[짧은 제목]{소 절 이름}
```

```
\chapter*{챕터 이름} % \documentclass[chapter]{oblivoir}
```

```
\section*{절 이름}
```

```
\subsection*{소 절 이름}
```

문서의 계층구조

```
\documentclass{oblivoir}
```

```
\title{라텍 글쓰기}
```

```
\author{권현우}
```

```
\date{2022년}
```

```
\begin{document}
```

```
\maketitle
```

```
\tableofcontents
```

```
\section{환영의 말}
```

안녕하세요.

```
\section{감사의 말}
```

감사합니다.

```
\section*{작별의 말}
```

만나서 반가웠습니다.

글자크기, 글자 강조하기

`\tiny` % 10pt 기준 6pt

`\scriptsize` % 10pt 기준 7pt

`\footnotesize` % 10pt 기준 8pt

`\small` % 10pt 기준 9pt

`\normalsize` % 10pt 기준 10pt

`\large` % 10pt 기준 10.95pt

`\Large` % 10pt 기준 12pt

`\LARGE` % 10pt 기준 14.4pt

`\huge` % 10pt 기준 17.28pt

`\Huge` % 10pt 기준 20.74pt

팍팍 팍팍 팍팍 팍팍 팍팍 팍팍 팍팍 팍팍 **팍팍 팍팍**

글자크기, 글자 강조하기

```
\tiny \scriptsize \footnotesize \small \normalsize
```

```
\large \Large \LARGE \huge \Huge
```

한번 명령어를 사용하면 global하게 적용되기 때문에 적용하고 싶은 곳에 그룹핑을 하거나 환경을 이용해야 한다.

예제

아래 문장에서 한 군데만 작게 하고 싶어요.

괜찮아요? {\tiny 작아서} 많이 놀랐죠?

```
\begin{footnotesize}
```

이 문단 전체를 작게 하고 싶어요.

작아졌지요?

```
\end{footnotesize}
```

글자크기, 글자 강조하기

`\textrm{...}`

Lectures on L^AT_EX (roman)

`\textsf{...}`

Lectures on L^AT_EX (sans serif)

`\texttt{...}`

Lectures on T_EX (typewriter)

`\textmd{...}`

Lectures on L^AT_EX (medium)

`\textbf{...}`

Lectures on L^AT_EX (bold face)

`\textup{...}`

Lectures on L^AT_EX (upright)

`\textit{...}`

Lectures on L^AT_EX (italic)

`\textsl{...}`

Lectures on L^AT_EX (slanted)

`\textsc{...}`

LECTURES ON L^AT_EX (small caps)

`\emph{...}`

Lectures on L^AT_EX (emphasized)

`\textnormal{...}`

Lectures on L^AT_EX (document font)

문단 정렬

```
\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

각주넣기

각주를 넣어봅시다. `\footnote{이건 각주!}`

- 각주 모양이 마음에 안들어요

각주넣기

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

```
\begin{document}
```

```
각주를 넣어봅시다.\footnote{이건 각주!}
```

```
\end{document}
```

예제 실습하기

목표

크기가 B5인 문서로 좌우여백 25mm, 상여백 23mm, 하여백 28mm으로 된 수학 소 논문을 과제로 제출할 수 있다.

본문서체 11pt

Step 1. 글의 구조

```
\documentclass[11pt,a4paper]{oblivoir}%9pt, 10pt, 10.5pt, 11pt, 12pt, 14pt, 17pt
%\documentclass[11pt,chapter,a4paper]{oblivoir}%twoside, openright, openleft, openany
%\documentclass[10.5pt,romanfixed]{oblivoir}
```

- chapter가 없는 문서 : **article**
- chapter가 있는 문서 : **book**

Step 1. 글의 구조

- part
- chapter
- section
- subsection
- subsubsection
- paragraph
- subparagraph

Q: 를 붙이면 어떻게 될까?

Step 1. 글의 구조

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

```
\counterwithin{section}{chapter}
```

```
\setsecnumdepth{subsection}%memoir, oblivoir
```

- 첫 번째 명령어는 절 번호에 장 번호가 나타나지 않도록 막아주는 명령어다.
- 두 번째 명령어는 새로운 장이 시작할 때마다 절 번호도 처음부터 다시 시작하도록 만들어주는 명령어다.
- 세 번째 명령어는 `subsection` 단위에도 번호를 매겨준다.

Step 2. 문서 양식 / 여백

%oblivoir

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

```
\usepackage{fapapersize}
```

```
\usefapapersize{210mm,297mm,30mm,*,30mm,32mm}
```

- 공급용지 : 인쇄될 용지
- 편집용지 : 재단해서 바라보는 용지

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

```
\usepackage{fapapersize}
```

```
\usefastocksize{210mm,297mm}
```

```
\usefapapersize{184mm,257mm,25mm,*,25mm,28mm} %좌측여백, 우측여백, 위측여백, 아래측여백
```

Step 2. 문서 양식 / 행간 조절

```
\SetHangulspace{1.3}{1.1} %\begin{document} 이전
```

```
\ResetHangulspace{1.3}{1.1} %문서 중간에서 행간을 조절
```

```
%not oblivoir
```

```
\usepackage{setspace}
```

```
\usepackage{footmisc}
```

```
\renewcommand{\footnotelayout}{\setstretch{1.1}}
```

```
\setstretch{1.3}
```

Step 2. 문서 양식 / 서체 설정

```
\usepackage{kotex}
```

```
\setmainfont{TeX Gyre Termes} %영문 세리프 폰트
```

```
\setsansfont{TeX Gyre Heros} %영문 산세리프 폰트
```

```
\setmonofont{Lucida Console} %영문 모노 폰트
```

```
\setmainhangulfont[BoldFont={KoPubBatang Medium}]{KoPubBatang Light} %한글 세리프 폰트
```

```
\setsanshangulfont[BoldFont={KoPubDotum Medium}]{KoPubDotum Light} %한글 산세리프 폰트
```

```
\setmonohangulfont{D2Coding} %한글 모노 폰트
```

```
\setmainhanjafont{HanaMinA} %한자 세리프 폰트
```

- 강제 기울임(FakeSlant), 강제 장평조정(FakeStretch), 강제 굵기(FakeBold) 모두 가능하다. 추천하진 않는다.
- 여러가지 서체 설정 옵션들이 있다. 자세한 것은 `fontspec` 메뉴얼 참조

Step 2. 문서 양식 / 서체 설정

참고로 Overleaf에서 서체를 적용하려면 서체파일을 업로드하고

```
\setmainhangulfont{NanumGothic}[  
  Path=./Fonts/,  
  Extension = .ttf,  
  UprightFont=*,  
  BoldFont=*-Bold,  
]
```

와 같은 종류의 명령어를 입력하면 된다.

Step 3. 각주, 기호문자

- 명령어는 `\footnote{각주}`
- 방주도 있다. `\marginpar{방주}`
- 문서의 상황에 따라 `\footnotemark`, `\footnotetext{각주내용}`을 사용해야 할 때도 있다.
- Oblivoir의 경우 한국식 각주를 쓰고 싶다면

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

다양한 각주를 쓰고 싶으면

```
\usepackage[옵션]{dhucsfn}
```

Step 4. 페이지 스타일

```
\makepagestyle{myheading}
\makeevenfoot{myheading}{\thepage}{}{}
\makeoddfoot{myheading}{}{}{\thepage}
\makeevenhead{myheading}{2022-1\quad 해석개론 I}{}{20220001 홍길동}
\makeoddhead{myheading}{20220001 홍길동}{}{1차 과제}

\pagestyle{myheading}
```

- \LaTeX 고유명령어로 고치는 것은 상당한 노고가 필요하다.
- 만약 memoir 또는 oblvioir를 사용할 경우 fancyhdr와 같은 패키지가 유용하다.
- 책 출판의 경우 고려해야 할 것이 더 있다. leftmark, rightmark, ...

Step 5. 참고 문헌

- 본문 인용 `\cite{}`
- 참고문헌 (thebibliography)

```
\begin{thebibliography}{라벨}
```

```
\bibitem[옵선키]{인용라벨}
```

```
\end{thebibliography}
```

참고문헌에 대한 이야기는 3강에서 더 자세히 다룬다.

Step 5. 참고 문헌

유계 립쉬쯔 영역에서 정상 나비에-스토크스 방정식의 약해의 존재성은 1933년에 장 르레
→ `\cite{J}`가 증명했다. 그 후, 1934년에 장 르레는 1934년에 `\cite{J2}`에서 비정상
→ 나비에-스토크스 방정식의 약해의 존재성을 증명했다.

```
\begin{thebibliography}{9}
```

```
\bibitem{J} J. Leray, \textit{Étude de diverses équations intégrales non linéaires et  
→ de quelques problèmes que pose l'hydrodynamique}, J. Math. Pures Appl. 12 (1933),  
→ 1--82.
```

```
\bibitem{J2} J. Leray, \textit{Sur le mouvement d'un liquide visqueux emplissant  
→ l'espace}, Acta Math. 63 (1934), no. 1, 193--248.
```

```
\end{thebibliography}
```

Step 6. AMS Environment

- 정리 환경 등을 표현하는 패키지
- 수학 논문에서는 표준
- 한글 문서에는 다소 적합하지 않아 다른 패키지들을 혼합해서 사용하는 것이 좋으나, 형식은 대부분 준수함.

```
\usepackage{amsthm}
```

```
\theoremstyle{plain}
```

```
\newtheorem{theorem}{Theorem}[section]
```

```
\theoremstyle{definition}
```

```
\newtheorem{definition}[theorem]{Definition}
```

```
\theoremstyle{remark}
```

```
\newtheorem{remark}[theorem]{Remark}
```

```
...
```

Step 6. AMS Environment

```
\begin{definition}
```

A natural number p is said to be a **prime** if it is greater than 1 and cannot be
→ written as the product of two smaller numbers.

```
\end{definition}
```

```
\begin{theorem}[Euclid]
```

There are infinitely many prime numbers.

```
\end{theorem}
```