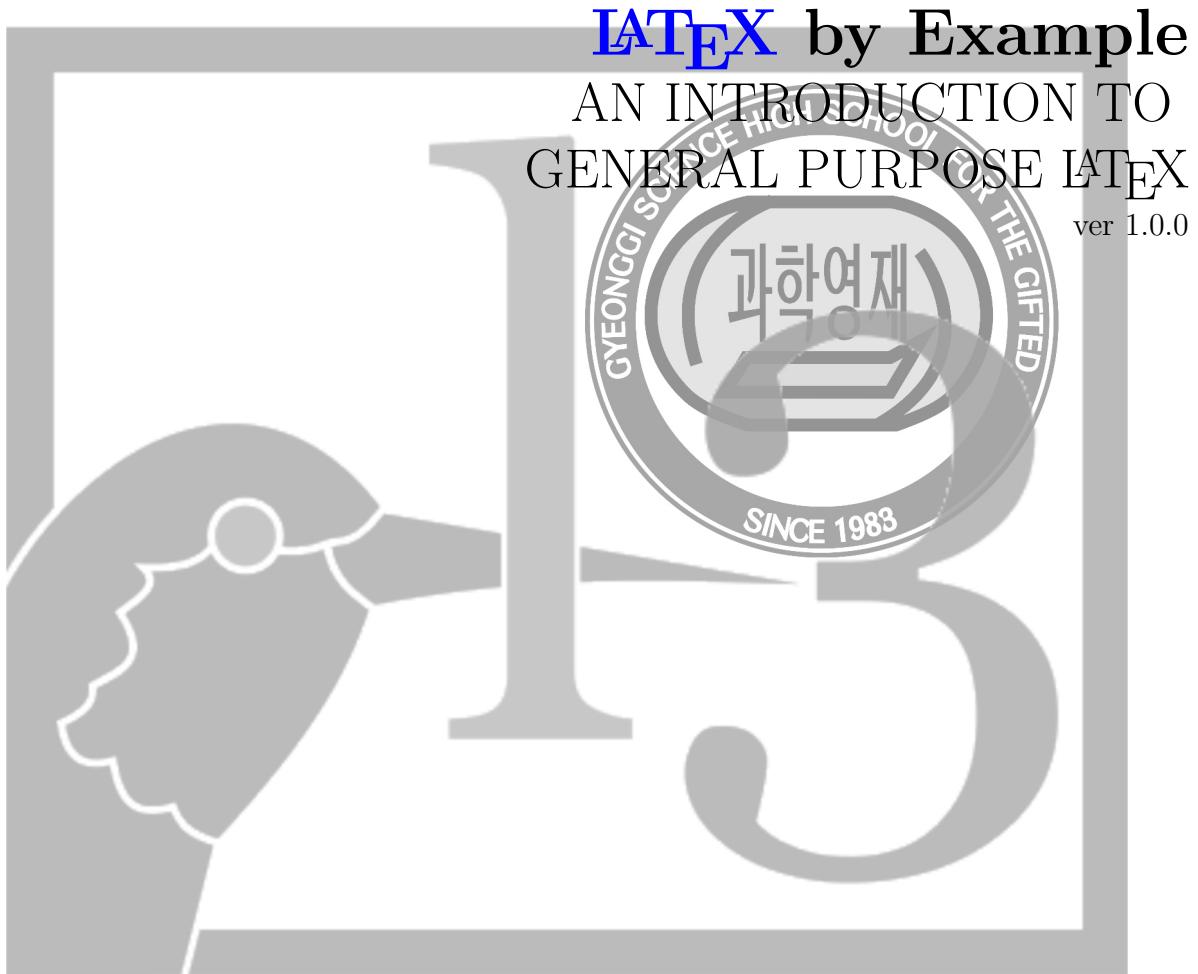


예제로 배우는 LATEX 문서 작성

입문자를 위한 LATEX 문서 작성의 기초



경기과학고등학교 LATEX 사용자 협회 지음

Contents

I	왜 L^AT_EX인가? L^AT_EX 시작하기	1
1	이번 장의 목표	1
2	L ^A T _E X 설치하기	1
2.1	kotexlive 설치	1
2.2	Editor 설치	1
3	L ^A T _E X란 무엇인가	1
4	L ^A T _E X를 사용하는 이유	2
II	L^AT_EX와 친해지기	3
1	이번장의 목표	3
2	L ^A T _E X 문서구조	3
2.1	Preamble(프림블)	3
2.2	Body(본문)	3
3	용어 설명	3
4	Hello World!!	4
4.1	굵게, 기울임꼴, 밑줄	4
4.2	글꼴 크기	5
4.3	특수 기호	5
4.4	내장 폰트	6
5	L ^A T _E X 단락 설정하기	7
5.1	이 단원의 목표	7
5.2	개행하기	7
5.3	좌측, 우측, 중간 정렬	7
5.4	아래정렬	8
5.5	줄간격 조절하기	8
5.6	주석, 메모, 코멘트	9
5.7	Section 나누기	9
III	수식에 특화된 L^AT_EX	10
1	이번 장의 목표	10
2	수식 입력하기	10
3	기호 입력하기	11
4	연산자 입력하기	12
5	엑센트 입력하기	13
IV	여러가지 요소삽입	14
1	이번장의 목표	14
2	목차 삽입하기	14
3	각주 삽입하기	14
4	아이템 열거 삽입하기	15
5	이미지 삽입하기	16

6 표 삽입하기	18
7 그래프 삽입	20
7.1 tikz	20
7.2 점	21
7.3 선	21
7.4 도형	22
7.5 함수	23
V Advanced Mathematics	24
1 수식 라벨링	24
1.1 수식 라벨링	24
1.2 수식 상호 참조	25
2 수식 정렬하기	25
3 Formatting Equations	26
3.1 size	26
3.2 color	27
3.3 Fonts	27
3.4 boxed	27
VI Advanced use of L^AT_EX	28
1 자동 조사	28
2 참고 문헌 삽입	28
3 referencing	29
3.1 labeling	29
3.2 referencing	29
4 algorithms	30
5 source code	30
VII 마치며	31

I. 왜 LATEX인가? LATEX 시작하기

1. 이번 장의 목표

이번 장을 통해 여러분은 다음 사항을 학습할 것이다:

- LATEX 설치하기
- LATEX란 무엇인가
- LATEX를 사용하는 이유

2. LATEX 설치하기

여러분은 이 책의 1단원 목차 순서를 보고 이상한 기분이 들 것이다. 왜냐하면, 주로 무엇에 대해 소개를 하는 책은 그것이 무엇인지와 그것을 사용하는 이유에 대하여 먼저 설명을 한 다음에 설치를 하는 방법에 대하여 설명하거나 아예 설치를 하는 방법이 아예 나와있지 않는 경우도 많기 때문이다. 하지만 설치방법이 먼저 나온데에는 깊은 뜻이 있다. 컴퓨터의 성능에 따라 LATEX를 설치하는데 1시간이 넘게 걸리는 경우도 있기 때문에 설치를 시작하고 LATEX에 대하여 알아보는 것을 권장한다.

2.1. kotexlive 설치

TeXLive는 TeX의 문서작성 시스템 외에 여러가지 TeX관련 프로그램, 패키지, 매크로, 등을 포함하고 있으며, TeX 사용 환경을 조성해주는데에 사용이 된다. 한글을 입력하기 위해서는 kotexlive가 필요하며 kotexlive는 아래 주소에서 다운 받을 수 있다.

www.ktug.org/xe/install

2.2. Editor 설치

TeX를 사용하기 위해서는 Editor가 필요하다. Microsoft Windows와 같은 경우에는 TeXworks, TeXstudio, TeXnicCenter와 같은 여러 무료 프로그램이 존재한다. TeX를 사용했던 경기과학고등학교 선배들은 TeXstudio 사용을 권장했다. TeXstudio는 아래 주소를 통해 다운 받을 수 있다.

www.texstudio.org/

3. LATEX란 무엇인가

자, 이제 드디어 여러분이 궁금한 LATEX가 무엇인지 알아보는 시간이다. LATEX는 텍스트 및 문서의 구조와 의미를 나타내는 명령어를 작성하고 LATEX프로그램을 통하여 깔끔한 형태의 문서를 만드는 것이다. 즉, LATEX는 마이크로소프트 워드와 같이 어떻게 보이는지 WYSIWYG(what you see is what you get)에 중점을 두지 않고 HTML와 같이 어떤 것인지 WYSIWYM(what you see is what you mean)에 중점을 둔다.

4. L^AT_EX를 사용하는 이유

L^AT_EX를 사용하는 데에는 여러가지 장점이 있다

- 수식편집기로 써는 수학에서 표준으로 쓰인다.
 - 초기 설정을 잘 해두면 작업량이 줄어들게 된다.
 - 차례, 그림/표 목차를 명령어 하나로 추가할 수 있다.
 - 참고문현을 자동으로 인용순으로 정렬 할 수 있다.
 - Cross-referencing 을 쉽게 할 수가 있다.
 - 벡터 이미지(svg, eps, pdf)를 손쉽게 첨부를 할 수가 있다.
- 하지만 어떠한 것도 장점만 있을 수는 없다. L^AT_EX의 단점으로는
- 처음 사용하는 사람이 읽거나 편집하기가 어렵다
 - 양식을 초기 설정을 하기가 어렵다
 - 실시간으로 편집을 하고 있는게 보이지 않는다

위의 단점에서 알 수 있듯이 L^AT_EX를 처음 사용하는 여러분은 초기설정이나 양식을 맞추는데에 어려움이 있을 수 있다. 이 때문에 경기과학고등학교 L^AT_EX사용자 협회가 설립 되었으며 여러분을 위해 경기과학고등학교에서 생활하면서 사용하게 될 L^AT_EX양식과 학습자료, 그리고 예시로 작성이 된 여러가지 문서들을 경기과학고등학교 L^AT_EX사용자 협회 홈페이지 (latex.gs.hs.kr)에서 제공한다(참고로 이 책도 L^AT_EX로 작성이 되었으며 모든 학습자료 역시 L^AT_EX로 작성이 되어 있다).

II. L^AT_EX와 친해지기

1. 이번장의 목표

이번 장에서 여러분은 다음 사항을 학습 할 것이다.

- L^AT_EX의 문서 구조
- L^AT_EX글꼴 설정
- L^AT_EEX단락 설정

2. L^AT_EEX문서구조

L^AT_EEX문서는 크게 두 부분으로 나누어 지게 된다

2.1. Preamble(프림블)

Preamble은 문서의 유형(documentclass) 들을 선언하며, 필요한 환경설정 및 명령어 역시 여기서 선언을 한다. 프로그래밍의 Header와 같은 역할을 하게 된다.

2.2. Body(본문)

본문에서는 예상을 할 수가 있듯이 문서의 내용, 문서의 글꼴과 같은 완성이 된 문서의 내용 및 모양을 작성하는 부분이다. 경기과학고등학교 L^AT_EEX사용자 협회 일반 부원이라면 주어진 양식으로 주로 작업을 하는 부분이 본문일 것이다.

이 책의 예시에서는 Preamble과 Body를 //... 로 구분을 해 놓았다

3. 용어 설명

이 책을 읽으면서 보게될 용어를 설명 해 주려 한다.

처음으로 **환경**이다. 환경은 환경안의 텍스트가 어떠한 의미를 가질지 정의하는데 사용이 된다. 예를 들어 \begin{asdf}와 \end{asdf}안에 있는 텍스트는 “asdf 환경 안에 있다”라고 표현한다. 환경에는 글꼴을 나타내는 환경, 표를 나타내는 환경, 단락을 나타내는 환경 등, 여러가지가 있다.

다음은 **패키지**이다. 패키지는 기본 제공되는 환경 외에 환경안의 스크립트가 어떠한 역할을 할지 알려주는 라이브러리와 같은 역할을 한다. 만약 여러분이 TeXLive를 통하여 테글 설치 했다면 이 책에서 다루는 패키지는 전부 설치가 되었을 것이다. 패키지는 프림블 \usepackage{sample}을 입력하여 사용 할 수 있다.

4. Hello World!!

LATEX를 사용하는 이유, 장점, 문서구조에 대하여 배우고 설치 까지 완료를 했으면 이젠 실제로 문서를 작성할 시간이 되었다. TeXEditor을 키고 새 문서를 만들어 보자. 우리는 이젠 코딩의 관례에 따라 Hello World!!가 쓰인 문서를 작성해 볼 것이다.

\documentclass{article} \begin{document} Hello World!! \end{document}	Hello World!!
--	---------------

이 책에서의 예시는 따로 언급이 없다면 왼쪽에는 LATEX코드를 오른쪽에는 컴파일 결과를 표시하겠다.

만약 한국어를 입력하고 싶다면 프림블에 ‘\usepackage{kotex}’를 입력하고 사용하면 된다.

여러분은 이제 Hello World!!가 적힌 문서를 LATEX를 통하여 만들 수 있게 되었다. 하지만 여러분은 이 Hello World!! 가 너무 평범해 보인다는 것에 동의를 할 것이다. 우리는 여러분의 기대를 저버리지 않기 위해 이제 Hello World!!가 다양한 글꼴을 가지게 할 것이다.

4.1. 굵게, 기울임꼴, 밑줄

아래는 글자를 *italic*, **bold**로 작성할 때나 밑줄치는 예시이다.

이 예제부터는 따로 패키지를 사용하지 않는 경우 본문 안의 코드만 표기 할 것이다

\textit{Hello World!!} \it Hello World!! }	<i>Hello World!!</i> <i>Hello World!!</i>
\textbf{Hello World!!} \bf Hello World!! }	Hello World!! Hello World!!
\underline{Hello World!!}	<u>Hello World!!</u>

눈치 빠른 여러분들은 알아 차렸듯이 italic이나 bold 체는 사용할 수 있는 방법이 두 가지가 있다. 여러분이 사용하기 편한대로 사용하면 된다.

4.2. 글꼴 크기

글자를 italic, bold 그리고 밑줄까지 쳤으면 이젠 문자의 크기를 조절 할 차례이다.

\tiny Hello World!!	Hello World!!
\scriptsize Hello World!!	Hello World!!
\footnotesize Hello World!!	Hello World!!
\small Hello World!!	Hello World!!
\normalsize Hello World!!	Hello World!!
\large Hello World!!	Hello World!!
\Large Hello World!!	Hello World!!
\LARGE Hello World!!	Hello World!!
\huge Hello World!!	Hello World!!
\Huge Hello World!!	Hello World!!

참고: 이 책은 12pt article 문서 클래스를 사용하고 있어 \huge와 \Huge의 크기 차이가 없다.

위와 같은 방법을 사용하면 \size 뒤에 있는 모든 텍스트 크기가 변하게 된다. 만약 이것처럼 특정 부분에만 글자 크기를 적용 하고 싶다면

{\LARGE LARGE} normal	LARGE normal
-----------------------	--------------

와 같이 사용하면 된다.

4.3. 특수 기호

이젠 특수부호를 입력하는 방법을 배울 시간이다. WYSIWYG인 워드 프로세서와 달리 LATEX와 같은 WYSIWYM인 문서를 작성할 때에는 많은 특수기호들이 명령어와 겹치기 때문에 일반적인 방법으로 특수 문자들을 입력을 해줄 수 없다. 몇 가지 특수기호들의 예시를 들어보겠다.

\%, \&, _, \\$, \#, \{, \}	\%, \&, _, \\$, \#, \{, \}
-----------------------------	-----------------------------

4.4. 내장 폰트

TeX는 다음과 같은 내장 폰트를 제공한다.

\emph{Emphasis}	<i>Emphasis</i>
\textrm{Roman Font Family}	Roman Font Family
\textsf{Sans Serif Font Family}	Sans Serif Font Family
\texttt{Teletypefont Family}	Teletypefont Family
\textsc{Small Capitals}	SMALL CAPITALS

TeX는 학술 논문 작성을 위해 만들어진 언어 이므로 이 이외의 폰트를 사용 할 일이 거의 없을 것이다. 하지만 만약 다른 폰트를 사용하고 싶으면 구글에 검색을 해보기를 추천한다.

축하한다! 여러분은 이젠 기본적인 문장을 L^AT_EX로 작성할 수 있게 되었다.

5. L^AT_EX 단락 설정하기

5.1. 이 단원의 목표

이번 장에서 여러분은 다음 사항을 학습 할 것이다.

- 개행하기
- 좌측, 우측, 중간 정렬
- 아래 정렬
- 줄간격 조절
- 주석, 메모, 코멘트
- Section 나누기

5.2. 개행하기

개행하는 방법에는 총 3가지가 있으며 2가지 결과가 나오게 된다.

Hello World!!	Hello World!!
Hello World!!	Hello World!! Hello World!!
Hello World!!\newline Hello World!!	Hello World!! Hello World!!
Hello World!! \\Hello World!!	Hello World!! Hello World!!

위에서 볼 수 있듯이 두 번 개행할 경우 새로운 문단으로 시작하게 되며 다른 두 방법은 줄만 바뀌게 된다.

5.3. 좌측, 우측, 중간 정렬

좌측, 우측, 중간 정렬은 적용할 글자가 적을 경우에는 다음과 같은 방법을 사용하면 된다.

{\flushleft Hello World!! }	Hello World!!
{\flushright Hello World!! }	Hello World!!
{\centering Hello World!! }	Hello World!!

적용할 글자가 한 문단이거나 많을 경우에는 다음과 같은 방법을 사용하는 것을 추천한다.

\begin{flushleft} Hello World!! \end{flushleft}	Hello World!!
\begin{flushright} Hello World!! \end{flushright}	Hello World!!
\begin{center} Hello World!! \end{center}	Hello World!!

5.4. 아래정렬

이 책의 표지와 같이 글을 페이지 아래에 적어야 할 때도 있다. 그럴 경우 아래와 같은 방법을 사용하면 된다.

\vfill Hello World!!	
	Hello World!!

5.5. 줄간격 조절하기

우리는 가독성을 높이기 위해서 줄간격을 조절 할 때가 있다. 여러분은 크게 2가지 방식으로 줄간격을 조절 할 수가 있다.

Contents...	The quick brown fox jumps over the lazy dog
\usepackage{setspace} //... \onehalfspacing Contents...	The quick brown fox jumps over the lazy dog
\usepackage{setspace} //... \doublespacing Contents...	The quick brown fox jumps over the lazy dog

위와 같이 사용 할 경우 \---spacing 뒤의 모든 텍스트들에 적용이 되므로 예를 들어 한 문단에만 적용을 하고 싶으면 다음과 같이 사용하면 된다.

Contents...	The quick brown fox jumps over the lazy dog
\usepackage{setspace} //... \{spacing\}{1.5} Contents... \{spacing\}	The quick brown fox jumps over the lazy dog

5.6. 주석, 메모, 코멘트

코드를 작성할 때 여러분은 주석 처리하거나, 또는 문서 작성할 때 메모나 코멘트를 해야 하는 경우가 있었을 것이다.

주석 처리를 하기 위해서는 다음과 같은 방법을 사용한다.

visible dog % invisible cat	visible dog
--------------------------------	-------------

많은 글을 주석 처리를 해야 할 경우에는 다음과 같은 방법을 사용하기를 권장한다.

\usepackage{verbatim} //... visible dog \begin{comment} invisible cat \end{comment}	visible dog
--	-------------

그 외에도 여러가지 주석처리나 메모기능이 있으며 만약 다른 방법을 사용하고 싶다면 인터넷에서 찾아보기 바란다.

5.7. Section 나누기

논문이나 보고서를 작성할 때 이 책처럼 문서를 장, 단원, 등으로 나누어야 할 때가 있다. LATEX를 사용하면 손쉽게 Section 나누기를 할 수가 있다. LATEX특성상 표 안에 예시 결과를 넣을 수 없기 때문에 2장의 Section을 예시로 하겠다.

```
\section{LATEX와 친해지기 }
\subsection{이번장의 목표}
\subsection{LATEX의 문서구조 }
\subsubsection{Preamble}
\subsubsection{Body}
```

와 같이 사용하면 2장의 2.2까지 Section으로 나눌 수가 있다. 내용은 Section 사이사이에 들어간다.

III. 수식에 특화된 LATEX

1. 이번 장의 목표

- 수식 입력하기
- 기호 입력하기
- 연산자 입력하기
- 액센트 입력

2. 수식 입력하기

LATEX로 논문이나 보고서를 작성 할 때 수식을 넣어야 하는 경우가 많을 것이다. 수식을 입력 할 때 다음과 같은 환경을 만들어 주어야 한다.

Equation: \((1+b=3)\)	Equation: $1 + b = 3$
Equation: \([1+b=3]\)	Equation: $1 + b = 3$
Equation: \\$1+b=3\\$	Equation: $1 + b = 3$
Equation: \$\$1+b=3\$\$	Equation: $1 + b = 3$
Equation: \begin{math}1+b=3\end{math}	Equation: $1 + b = 3$
Equation: \begin{displaymath}1+b=3\end{displaymath}	Equation: $1 + b = 3$

홀수번째와 짝수번째의 차이는 수식을 텍스트와 같이 있느냐 아니면 따로 디스플레이 하느냐에 있다.

주의: \$\$...\$\$는 AMS-LATEX같은 매크로와 충돌이 생겨 문제가 생길 수 있으므로 사용을
지양해야 한다.

3. 기호 입력하기

수학에서는 여러가지 기호가 사용이 된다. 여러분은 그 여러가지 기호를 수식에도 입력을 해야 하기 때문에 이 단원에는 여러가지 기호의 예시를 보여 주도록 하겠다.

처음으로 바로 사용할 수 있는 기호목록이다.

이 장의 예제는 이제부터 예시는 수식 환경 안에 있는 내용만 표기 하겠다.

+ - / =!()[] <> ' :

그리스 문자들은 다음과 같이 사용하면 된다.

\alpha, \beta, \gamma, \pi, \phi, \varphi	$\alpha\beta\gamma\pi\phi\varphi$
---	-----------------------------------

다음은 삼각함수 사용의 예시이다.

수식	코드	수식	코드	수식	코드	수식	코드
sin	\sin	cos	\cos	tan	\tan	cot	\cot
arcsin	\arcsin	arccos	\arccos	arctan	\arctan		
sinh	\sinh	cosh	\cosh	tanh	\tanh	coth	\coth
sec	\sec	csc	\csc				

다음은 논리 기호 입력 방법이다.

기호	코드	기호	코드	기호	코드	기호	코드
\exists	\exists	\nexists	\nexists	\forall	\forall		
\subset	\subset	\supset	\supset	\in	\in	\ni	\ni
\notin	\notin	\emptyset	\emptyset	\top	\top	\bot	\bot
\rightarrow	\rightarrow	\leftarrow	\leftarrow	\Rightarrow	\Rightarrow	\Leftarrow	\Leftarrow
\leftrightarrow	\leftrightarrow	\Leftrightarrow	\Leftrightarrow	\wedge	\wedge	\vee	\vee

다음은 관계기호 입력 방법이다.

기호	코드	기호	코드	기호	코드	기호	코드
\leq	\leq	\geq	\geq	\ll	\ll	\gg	\gg
\subset	\subset	\supset	\supset	\subseteq	\subseteq	\supseteq	\supseteq
$\not\subseteq$	\not\subseteq	$\not\supset$	\not\supset	\parallel	\parallel	\nparallel	\nparallel
\prec	\prec	\succ	\succ	\preceq	\preceq	\succeq	\succeq
\doteq	\doteq	\equiv	\equiv	\approx	\approx	\cong	\cong
\sim	\sim	\simeq	\simeq	\neq	\neq	\propto	\propto

여기에 없는 다른 기호들은 인터넷에서 찾아보거나 TeXstudio를 사용할 경우 원쪽 바에 수식 아카이브가 있으므로 여러분은 유용하게 사용하기만 하면 된다.

4. 연산자 입력하기

여러분은 LATEX에서 여러가지 수학 기호를 배웠다. 이젠 배운 수학기호들을 사용 할 때가 왔다. 다음은 연산자를 이용한 수식입력의 예시이다.

$k_1+k^2=3$	$k_1 + k^2 = 3$
$\lim_{x \rightarrow \infty} x = \infty$	$\lim_{x \rightarrow \infty} x = \infty$
$\exp(a) = 1$	$\exp(a) = 1$
$\cos(3\theta) = \cos^3 \theta$	$\cos(3\theta) = \cos^3 \theta$
$x \equiv 3 \pmod{4}$	$x \equiv 3 \pmod{4}$
$\frac{a}{4}$	$\frac{a}{4}$
$\sum_{k=0}^{10} x$	$\sum_{k=0}^{10} x$
$\int_0^\infty x \, dx$	$\int_0^\infty x \, dx$
$\frac{d}{dx}(kg(x))$	$\frac{d}{dx}(kg(x))$
$\big(k f(x) \big)$	$\big(k f(x) \big)$
$x \in [-2, 1]$	$x \in [-2, 1]$
$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$

여기에 없는 다른 수식/연산자 들은 인터넷에서 찾아보자.

5. 엑센트 입력하기

수식을 입력할 때에 벡터와 같이 엑센트를 사용 하는 경우가 적지 않게 있다. 엑센트를 사용하는 예시는 다음과 같다.

a'	a'	a''	a''
\hat{a}	$\backslash\hat{a}\{a\}$	\bar{a}	$\backslash\bar{a}\{a\}$
\dot{a}	$\backslash\dot{a}\{a\}$	\ddot{a}	$\backslash\ddot{a}\{a\}$
\overrightarrow{AB}	$\backslash\overrightarrow{AB}\{AB\}$	\overleftarrow{AB}	$\backslash\overleftarrow{AB}\{AB\}$
\overline{abc}	$\backslash\overline{abc}\{abc\}$	\vec{v}	$\backslash\vec{v}\{v\}$

IV. 여러가지 요소삽입

1. 이번장의 목표

- 목차 삽입하기
- 각주 삽입하기
- 아이템 열거 삽입하기
- 이미지 삽입하기
- 표 삽입하기
- 그래프 삽입하기

2. 목차 삽입하기

책이나 논문은 많은 section으로 나워어 지는 경우가 많으며 이 section의 페이지나 대략적인 내용을 알려주기 위하여 목차를 넣는 경우가 많다. LATEX에서는 간단하게 한 스크립트만 입력을 해주면 목차를 작성해 주게 된다.

```
\tableofcontents
```

주의: 목차는 두번 컴파일 해 주어야 업데이트가 된다.

다음은 그림, 표 목록과 캡션을 출력하기 위해서는 다음 스크립트를 입력해 주면 된다.

```
\listoffigures  
\listoftables
```

3. 각주 삽입하기

각주 입력하기는 매우 쉽다.

Where is footnote?\footnote{Here it is!}	Where is footnote? ¹
--	---------------------------------

각주 입력하기는 매우 쉽다. 하지만 여러분은 각주의 위치에 주의해야 한다. 각주는 마지막 텍스트 다음 줄에 위치하게 된다. 그럼으로 만약 각주가 페이지 가장 밑에 있기를 원한다면 그 페이지 마지막 문장 다음에 \clearpage를 해주는 것이 좋다.

¹Here it is!

4. 아이템 열거 삽입하기

매장 시작마다 여러분은 단원을 나열한 리스트를 볼 수가 있었을 것이다. 이와 같은 아이템 리스트 들은 다음과 같이 서식 할 수가 있다.

```
\begin{itemize}
    \item Chocola
    \item Vanilla
    \item Maple
\end{itemize}
```

- Chocola
- Vanilla
- Maple

만약 번호가 있는 열거를 원한다면 enumerate환경을 사용하면 된다.

```
\begin{enumerate}
    \item Chocola
    \item Vanilla
    \item Maple
\end{enumerate}
```

1. Chocola
2. Vanilla
3. Maple

5. 이미지 삽입하기

이미지는 다음과 같이 삽입하면 된다.

표 안에는 이미지를 추가 할 수 없으므로 이미지는 표 바로 아래 추가한다.

```
\begin{figure}[h]
    \centering
    \includegraphics[scale=1]{image_name.png}
\end{figure}
```



이미지에 캡션을 달기 위해서는 간단히 스크립트 한 줄만 더 적어주면 된다.

```
\begin{figure}[h]
    \centering
    \includegraphics[scale=1]{image_name.png}
    \caption{Example Image duke}
\end{figure}
```



Figure 1: Example Image duke

위 두 예시를 보고 여러분은 왜 [h]가 무슨 기능을 하는지 궁금했을 것이다. 대괄호([])안에 무엇이 들어가느냐에 따라서 이미지가 삽입되는 위치가 변하게 된다. 자세한 내용은 아래를 참조 바란다.

h	(here) 개체를 코드 위치에 삽입
t	(top) 개체를 페이지 맨 위에 삽입
b	(bottom) 개체를 페이지 맨 아래에 삽입
htbp	개체를 코드 위치에 우선적으로 삽입하고 페이지 경계를 넘어가거나 할 경우에 t,b,p와 같은 차선책을 사용하여 삽입

두단으로 나뉘어진 양식을 채택한 문서나 학술지 같은 경우 이미지를 중간에 삽입하기보다 위, 안 될 경우 아래에 삽입하는 것을 추천한다.

삽입할 사진의 유형에 따라 적합한 이미지 양식은 다음과 같다.

- 사진: jpg 또는 jpeg
- 그래프: pdf 또는 eps
- 기하적 그림: png 또는 벡터 이미지
- 일러스트: pdf (PowerPoint에서 ‘pdf로 내보내기’)

6. 표 삽입하기

여러분은 이 책을 보면서 수 많은 표를 보았을 것이다. 이젠 여러분이 직접 표를 만들 수 있게 표 만드는 방법을 가르쳐 주겠다. 맨 처음으로 표 제작은 **tabular**환경을 사용한다.

```
\begin{tabular}{|l||c|r|}\hline name&gender&offshore cash\\\hline\hline Trevor&male&200,000,000$\\\hline Michael&male&17,000,000$\\\hline\end{tabular}
```

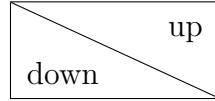
name	gender	offshore cash
Trevor	male	200,000,000\$
Michael	male	17,000,000\$

이젠 위 스크립트를 하나씩 알아볼 시간이다. { | | | c | r |}은 각 열을 좌측, 중간, 우측 정렬 시켜주며, 사이의 “|”는 수직선을 나타낸며 여러번 사용이 가능하고, \hline는 수평선을 나타내며 역시 여러번 사용 가능하다. 열 구분 기호로는 &이 사용이 되며, 행 구분 기호로는 \\이 사용이 된다.

다음은 여러가지 그래프를 그리는 예시이다.

이번 표 예시부터는 **tabular**환경안에 있는 스크립트만 서술하겠다.

```
\hline\n Diagbox{down}{up}\\\hline
```



```
\hline\n short&short&short\\\hline\n multicolumn{3}{|c|}{long}\\\hline
```

short	short	short

```
\hline  
short&short&short\\  
\hline  
short&\multicolumn{2}{c|}{long}\\  
\hline
```

short	short	short
short		long

주의: 위 예제와 같은 경우에는 “c|” 대신 “|c|”을 사용 할 경우 수직선이 중복되어 short와 long사이의 수직선만 두겹게 표기가 된다.

```
normal  
\hline \hline  
\multicolumn{1}{|l |}{left}\\  
\hline  
\multicolumn{1}{|r |}{right}\\  
\hline  
\multicolumn{1}{|c |}{center}\\  
\hline
```

normal
left
right
center

7. 그래프 삽입

7.1. tikz

tikz를 사용하여 LATEX에서 그래프 및 도형을 그릴 수가 있다. tikz를 사용하기 위해서는 tikz 패키지를 추가 해야 한다.

tikz를 사용 할 때 다음 사항을 주의 해야 한다.

- 데카르트 좌표계나 극 좌표계가 사용이 된다.
- 오른쪽으로 x좌표가 증가하고 위쪽으로 y좌표가 증가한다.
- 중심이 (0,0)이 아니라 상황에 따라 알맞은 위치로 맞춰진다.

tikz사용을 위해서는 다음과 같은 환경을 만드면 된다.

```
\usepackage{tikz}  
//...  
\begin{tikzpicture}  
    %scripts  
\end{tikzpicture}
```

또는

```
\usepackage{tikz}  
//...  
\tikz{ %scripts }
```

으로 사용해도 된다.

7.2. 점

tikz로 점을 입력할 때는 점은 나오지 않고 텍스트만 나오게 된다.

여러분도 이젠 예상할 수 있듯이 다음 예제 부터는 tikz환경 안에 있는 내용만 서술 할 것이다.

\node (a) at (0,0){a}; \node (b) at (-2,2){b};	b a
\path (0,0) node(x) {x} (2,2) node(y) {y}	y x

7.3. 선

선분을 그리는 방법은 다음과 같다.

\draw[option] (0,0) - - (1,1) - - (2,0);	
--	--

두 점을 직선이 아닌 택시 거리로 연결 할 수도 있다.

\draw (0,0) - (1,1);	
\draw (0,0) - (1,1);	

다음 같은 방법으로 베지 곡선도 만들 수 있다.

\draw (0,0) .. controls (1,1) .. (2,0);	
\draw (0,0) .. controls (1,1) and (2,1) .. (3,0);	

화살표도 간단하게 만들 수 있다.

<code>\draw [->] (0,0) - - (1,1) - -(3,0);</code>	
<code>\draw [<->] (0,0) .. controls(1,1) ..(3,0);</code>	

굵거나 점선으로 되어 있는 선분도 만들 수 있다.

<code>\draw[thick] (0,0) .. controls (1,1) .. (2,0);</code>	
<code>\draw[dashed] (0,0) .. controls (1,1) .. (2,0);</code>	

7.4. 도형

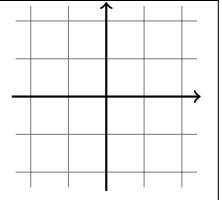
tikz를 사용하여 직선 이외에 다양한 도형을 그릴 수 있다. 이 때 `cycle`를 사용하면 첫번째 점으로 이여진다.

<code>\draw[thick] (0,0) - - (1,1) - - (2,0)- -cycle;</code>	
<code>\filldraw[blue] (0,0) - - (1,1) - - (2,0)- -cycle;</code>	
<code>\draw(0,0) circle[radius=5mm];</code>	
<code>\draw(0,0)</code> <code>circle[x radius=10mm, y radius=5mm];</code>	
<code>\draw (0 ,0) arc (150:30:5mm);</code>	
<code>\filldraw(0,0) - - (12 mm,0 mm)</code> <code>arc(0:30:12 mm) - - cycle;</code>	

7.5. 함수

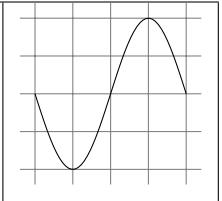
좌표계를 그리는 방법은 다음과 같다.

```
\draw[step =5mm, gray, thin](-1.2, -1.2) grid(1.2, 1.2);
\draw[->, thick ](-1.25, 0)- -(1.25, 0);
\draw[->, thick ](0, -1.25)- -(0, 1.25);
```



함수 그래프를 입력하기 위해서는 다음과 같이 정의역을 정해주고 사용하면 된다.

```
\draw[step=5mm, gray, thin ](-1.2, -1.2) grid(1.2, 1.2);
\draw[domain=-1:1, samples=50] plot(\x, sin(pi*\x r ));
```



여기서 “ $\pi * \x$ ” 뒤의 ‘r’은 radian을 의미한다.

V. Advanced Mathematics

1. 수식 라벨링

수학 논문의 경우 많은 수식이 사용되며 이것을 모두 번호를 매기고 참조를 하는데에 애를 먹을 수 있다. 심지어 중간에 수식을 첨부할 경우 뒤의 넘버링을 모두 바꾸어야 하고 상호 참조한 부분 까지 모두 일일이 찾아서 바꿔줘야 할 수도 있다. 하지만 L^AT_EX의 라벨링 기능을 사용하면 자동으로 상호참조 된 부분을 바꿔준다.

1.1. 수식 라벨링

equation환경을 사용하면 자동으로 라벨링을 해준다.

equation: $\begin{aligned} \begin{array}{l} \text{equation:} \\ \backslash begin\{equation\} \\ f(x) = (x+1)(x+2) \\ \backslash end\{equation\} \end{array} \end{aligned}$	equation: $f(x) = (x + 1)(x + 2) \quad (1)$
---	--

amsmath패키지를 사용하면 단원별로 라벨링이 가능하다.

$\begin{aligned} \begin{array}{l} \text{\usepackage\{amsmath\}} \\ \text{\numberwithin\{equation\}\{subsection\}} \\ //... \\ (\text{equation script...}) \end{array} \end{aligned}$
--

종속방정식을 라벨링하기 위해서는 subequations환경을 사용하면 된다.

$\begin{aligned} \begin{array}{l} \text{\begin\{subequations\}} \\ \text{\begin\{align\}} \\ 1+1&=2\\ 2&=1+1 \\ \text{\end\{align\}} \\ \text{\end\{subequations\}} \end{array} \end{aligned}$
--

$$1 + 1 = 2 \quad (2a)$$

$$2 = 1 + 1 \quad (2b)$$

1.2. 수식 상호 참조

\label와 \ref를 사용하면 손 쉽게 레퍼런싱을 할 수가 있다.

```
\begin{equation}
    \label{eq:sample}
    f(x) = x+1
\end{equation}
you can reference \ref{eq:sample} like this
```

$$f(x) = x + 1 \quad (3)$$

you can reference 3 like this

또는 다음과 같이 레퍼런싱 할 수 있다.

or you can reference like \ref{eq:sample}

or you can reference like (3)

2. 수식 정렬하기

LATEX는 깔끔한 논문을 작성 할 수 있도록 도와준다. 그럼 역시 수식을 깔끔하게 정렬하는 방법도 당연히 있을 것이다. 이번 단원에서는 여러문은 (2)와 같이 수식을 정렬하는 방법을 배울 것이다.

수식 정렬은 amsmath패키지의 align 또는 align*(수식 라벨링이 안됨)을 사용한다.
align 환경은 표 안에서 사용이 불가능 함으로 결과는 표 아래에 표시 하겠다.

```
\begin{align*}
    f(x) &= (x+1)(x+2) \\
    &= x^2 + 3x + 2
\end{align*}
```

$$\begin{aligned}
 f(x) &= (x + 1)(x + 2) \\
 &= x^2 + 3x + 2
\end{aligned}$$

위 예시에서 알 수 있듯이 \\는 줄을 구분할 때에 사용이 되고 &은 맞출 위치를 지정 할 때에 쓰인다.

align은 다음과 같아도 사용 할 수가 있다.

```
\begin{align*}
    f(x) &= 1+2+3 \nonumber \\
    &\quad &+4+5
\end{align*}
```

$$\begin{aligned}
 f(x) &= 11 + 2 + 3 \\
 &\quad + 4 + 5
\end{aligned}$$

다음과 같은 더 복잡한 수식의 정렬도 가능하다.

```
\begin{align*}
f(x) &= x^2 & g(x) &= 2 \\
f'(x) &= 2x & g'(x) &= 0
\end{align*}
```

$$\begin{array}{ll} f(x) = x^2 & g(x) = 2 \\ f'(x) = 2x & g'(x) = 0 \end{array}$$

`cases`환경을 사용하면 다음과 같은 수식 정렬도 가능하다.

<pre>\(f(x)= \begin{cases} x^2+3x+2 & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases} \)</pre>	$f(x) = \begin{cases} x^2 + 3x + 2 & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases}$
--	---

3. Formatting Equations

여러분은 문서를 작성 할 때에 수식을 강조를 할 때가 있을 것이다. LATEX에서는 수식에 여러 가지 색, 폰트, 크기를 적용 할 수 있다.

3.1. size

수식 환경안에서 크기 조절을 하는 스트립트는 다음과 같다.

<code>\displaystyle asdf</code>	<i>asdf</i>
<code>\textstyle asdf</code>	<i>asdf</i>
<code>\scriptstyle asdf</code>	<i>asdf</i>
<code>\scriptscriptstyle asdf</code>	<i>asdf</i>

3.2. color

수식에 색을 넣고 싶다면 **xcolor** 패키지를 이용하여 다음과 같이 사용하면 된다.

$f(\{\color{red}\}x)=x-2$	$f(\textcolor{red}{x}) = x - 2$
---------------------------	---------------------------------

3.3. Fonts

수식에는 기본으로 제공이 되는 다양한 폰트가 존재한다. 마지막 두 폰트를 제외하고 별다른 패키지 없이 사용 할 수 있다.

<code>\mathnormal{asdf1234}</code>	<i>asdf1234</i>
<code>\mathrm{asdf1234}</code>	<i>asdf1234</i>
<code>\it{asdf1234}</code>	<i>asdf1234</i>
<code>\bf{asdf1234}</code>	asdf1234
<code>\sf{asdf1234}</code>	<i>asdf1234</i>
<code>\tt{asdf1234}</code>	<i>asdf1234</i>
<code>\frak{asdf1234}</code>	<i>asdf1234</i>
<code>\cal{asdf1234}</code>	<i>ASDF</i>
<code>\bb{asdf1234}</code>	<i>ASDF</i>
<code>\scr{asdf1234}</code>	<i>ASDF</i>

위의 예시에서 볼 수가 있듯이 밑 3 예시는 영문 대문자만 가능한 폰트이며, 마지막 두 폰트는 순서대로 **amafonts**와 **mathrsfs** 패키지를 사용하면 된다.

3.4. boxed

수식에 테두리를 적용하는 방법은 수식 환경 내에서 다음과 같이 스크립트를 입력하면 된다.

<code>\boxed{x^2+y^2=4}</code>	$x^2 + y^2 = 4$
--------------------------------	-----------------

%**align**환경 안에서는 `\Aboxed{}`를 사용하면 된다.

VI. Advanced use of L^AT_EX

1. 자동 조사

L^AT_EX에는 자동으로 상호참조 해주는 스크립트가 존재한다. 하지만 한국어로 문서를 작성해야 하는 경우 “1과..., 2와...”같이 숫자에 따라 조사가 다르게 쓰이는 경우가 있다. 여러분은 이럴 때에 다음과 같은 자동 조사기능을 사용하면 추후에 조사를 일일이 검사 해야 하는 문제를 없앨 수 있다.

자동 조사 명령을 다음과 같다.

```
\이 \가, \을 \를, \와 \과, \로 \으로, \은 \는, \라 \이라
```

2. 참고 문헌 삽입

참고문헌을 삽입하는 방법은 `thebibliography`환경에 다음과 같이 작성하면 인용순 정열을 해준다.

```
\begin{thebibliography}{99}
    \bibitem{example1} Authors, “Article name”, Journal name, book, number, page, (year).
```

필요한 부분에 인용 표기를 하고자 한다면 해당 위치에 다음 스크립트를 입력하면 된다.

```
\cite{example1}
```

`bibitem{}`안의 내용을 기억하기 쉽게, 예로 주저자의 last name과 출판 년도를 사용하면 표기 를 한다. (ex: `bibitem{choi17}`)

3. referencing

3.1. labeling

여러분은 다음과 같은 환경에 레이블을 달아 다음에 레퍼런싱을 할 때 사용할 수 있다.

- 섹션
- 이미지, 표
- 수식
- 그 외에 캡션을 달수 있는 환경

수식 라벨링 상호 참조는 V장에서 이미 다루었다. 하지만, 나머지 요소도 가능하다는 것을 알려주기 위하여 이 단원을 만들었다.

위와 같이 색션에는 라벨링을 하기 위해서는 색션 내에 아무곳이나 다음 스크립트를 입력하면 된다. 하지만 나중의 편의를 위해서는 색션 바로 다음에 레이블을 추가하는 것을 권장한다.

```
\label{sec:label_sample}
```

나중에 레퍼런싱 할 때 혼동을 최소화 하기 위하여 레이블 앞에 “sec:”와 같이 어떠한 것을 레이블 한 것인지 명시 해주는 것을 추천한다.

그 외에 이미지나 표에 레퍼런싱을 하기 위해서는 무조건 \caption{} 다음 또는 안에 위와 같이 레이블을 입력하면 된다.

3.2. referencing

레퍼런싱은 상호참조를 하고자 하는 위치에 다음 스크립트를 입력하면 된다.

```
\ref{sam:sample}
```

4. algorithms

정보와 같은 과목의 시행절차는 다음과 같이 알고리즘 시리즈로 나열하면 전하고자 하는 알고리즘을 효과적으로 전달을 할 수가 있다.

```
\begin{algorithm}[H]
\KwData{Learnig \TeX}
\KwResult{Learned \TeX}
\While{using \TeX}{%
    use what you know\;
    \eIf{you don't know sth}{%
        \eIf{It is described in this book}{%
            {Learn from this book\;}}
            {Ask Google\;}}
        }{keep going\;}
    }
\end{algorithm}
```

Data: Learnig \TeX
Result: Learned \TeX
while *using \TeX do*
 use what you know;
 if *you don't know sth then*
 if *It is described in this book*
 then
 | Learn from this book;
 else
 | Ask Google;
 end
 else
 | keep going;
 end
end

여기에서는 알고리즘 입력 외에 또 더 중요한 것을 다루고 있다. \TeX를 사용하면서 모르는 것이 나오는 것은 당연하다. 명령어를 모를 수 있고, 또는 컴파일을 할 때에 나오는 에러가 무엇인지 모를 수도 있다. 이때는 우리의 친구 Google에게 물어보도록 하자.

5. source code

우리의 긴 여행의 마지막 \TeX에서 소스 코드 입력하기이다. 소스코드를 입력하는데에는 두 가지 방법이 있으며 `listings` 패키지를 사용한다. 그 중 첫번째 예시는 다음과 같다.

```
\begin{tabularx}{\textwidth}{|X|X|}
```

```
\begin{tabularx}{\textwidth}{|X|X|}
```

만약 코드 파일이 있다면 다음과 같은 방법을 이용하는 것을 추천한다.

```
\lstinputlisting[language=TeX, firstline=1, lastline=1398]{example_source.py}
```

이것을 지원하는 언어 목록들은 다음과 같다:

ABAP, ACSL, Ada, Algol, Ant, Assembler, Awk, bash, Basic, C, C++, C, Caml, Clean, Cobol, Comal, csh, Delphi, Eiffe, Elan, erlang, Euphoria, Fortran, GCL, Gnuplot, Haskell, HTML, IDL, inform, Java JVMIS, ksh, Lisp, Logo, Lua, make, Mathematica, Matlab, Mercury, MetaPost, Miranda, Mizar, ML, Modelica, Modula-2, MuPAD, NASTRAN, Oberon-2, Objective C , OCL, Octave, Oz, Pascal, Perl, PHP, PL/I, Plasm, POV, Prolog, Promela, Python, R, Reduce, Rexx, RSL, Ruby, S, SAS, Scilab, sh, SHELL, Simula, SQL, tcl, TeX, VBScript, Verilog, VHDL, VRML, XML, XSLT.

VII. 마치며

이 책은 여러분이 TeX스크립트와 pdf 사이를 왔다갔다 하면서 공부 또는 작업하기보다, 왼쪽에 코드, 오른쪽에 결과로 예시를 작성하여 이 책만으로도 TeX공부를 할 수 있게 하고자 제작이 되었다. 책의 내용을 추가하는 것은 언제나 환영이다.