

LaTeX 소개 2부 : 수학, BibTeX, 사용자 정의 공개 통계학 개론 (OpenIntro Stat.) 저작 학습용

David Diez ¹ 이광춘(번역) ²

¹OpenIntro openintro.org

²xwMOOC xwmooc.net

목차

- \LaTeX 수학
- BibTeX: \LaTeX 참고문헌
- 사용자 정의 명령어와 환경 만들기
- 기타 비법

Guide to LaTeX

Guide to LaTeX 책에는 \LaTeX 에 대한 멋진 안내가 나와 있고, 이번 학습에서 이 책에서 나온 예제 일부를 충실히 따라간다:

7 수학

11,12 BibTeX

10 사용자 정의 명령어와 환경

\LaTeX 에 관한 학습교재를 찾는다면, *Guide to LaTeX* 책은 훌륭한 대안이 될 수 있다.

LaTeX수학

LaTeX에서 제공하는 수학 환경 몇가지 측면을 다룰 예정이다.

- 텍스트에 수식 기초
- 다양한 방정식 환경
- 수학 기호
- 수학 표현식
- 텍스트 강조와 변형
- 괄호 기호의 자동 크기 조정
- 수학 방정식에 텍스트
- 배열과 행렬

수식을 텍스트에 삽입

LaTeX에서 α , ζ , μ ... 같은 그리스 문자를 추가하기 쉽다. 동일한 방식으로 방정식도 쉽게 추가될 수 있다: $y = x^3$, $\sum z^j$,

$$x_1 + \cdots + x_n.$$

LaTeX makes it easy to add Greek letters like `α`, `ζ`, `μ`, etc. into text. In the same way, equations can be added easily as well: `$y=x^3$`, `$\sum z^j$`, `$x_1+\cdots+x_n$`.

`$` 기호는 LaTeX으로 하여금 언제 수식 모형으로 들어가거나 빠져나올지를 알려준다. 예를 들어, 상기 α 를 생성하려면, `α` 타이핑한다.

β 을 어떻게 생성할 수 있을까요?

방정식 배열

방정식 일부는 길어서 텍스트와 함께 표기되지 못해 그 차제로
행에 표시되어야 한다. 이런 경우에 `eqnarray` 혹은 `eqnarray*`
환경을 사용한다:

```
\begin{eqnarray*}
\sum_{k=0}^{\infty} 0.5^k = \frac{1}{1-0.5} = 2
\end{eqnarray*}
```

`eqnarray*`으로 \LaTeX 표현 결과::

$$\sum_{k=0}^{\infty} 0.5^k = \frac{1}{1-0.5} = 2$$

방정식 참조

표와 그림과 마찬가지로, 방정식을 참조할 수 있다. `eqnarray` (별표 없음) 을 사용해서 방정식 번호를 붙인다:

$$\sum_{k=0}^{\infty} 0.5^k = \frac{1}{1-0.5} = 2 \quad (1)$$

`\label{powerSeries}` 라벨을 방정식 배열 내부에 두고 나서,
`\ref{powerSeries}` 을 통해 참조한다.

```
\begin{eqnarray}
\sum_{k=0}^{\infty} 0.5^k = \frac{1}{1-0.5} = 2
\label{powerSeries}
\end{eqnarray}
```

줄맞춘 방정식

또다른 환경, `align` (그리고 `align*`) 은 여러행에 걸친 방정식 줄을 맞추는데 편리하다.

```
\begin{align}
(a+b)^3 &= (a+b) (a^2 + 2ab + b^2) \notag \\
&= a^3 + 3a^2b + 3ab^2 + b^3 \\
\end{align}
```

결과는 다음과 같다:

$$\begin{aligned} (a+b)^3 &= (a+b)(a^2 + 2ab + b^2) \\ &= a^3 + 3a^2b + 3ab^2 + b^3 \end{aligned} \quad (2)$$

`\\` 명령어로 줄바꿈한다. `\notag` 명령어로 첫행에 방정식 숫자가 나타나지 못하게 한다. 여기에, `amsmath` 패키지가 필요하다. (질문: 방정식 번호가 있으려면, 코드에 무엇을 포함해야 할까?)

다수 줄맞춤

align 환경은 다수 정렬을 가능케 한다:

```
\begin{align*}
(a+b)^0 &= 1 && & (a+b)^1 &= a+b \\
(a+b)^2 &= a^2 + 2ab + b^2 && & (a+b)^3 &= a^3 + 3a^2b + 3ab^2 + b^3
\end{align*}
```

출력 결과는 다음과 같다.

$$(a + b)^0 = 1$$

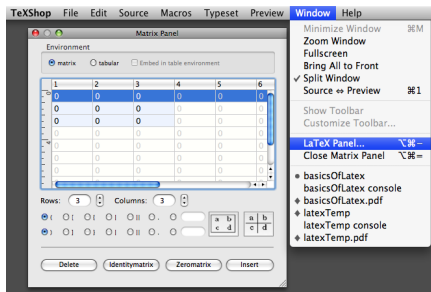
$$(a + b)^1 = a + b$$

$$(a + b)^2 = a^2 + 2ab + b^2$$

$$(a+b)^2 = a^2 + 2ab + b^2 \quad (a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

수학과 기호

모든 수학 구문을 배우는 것은 어렵다. \LaTeX 과 Matrix Panels이 도움이 많이 된다:



Matrix Panel 이 특히 유용한데 배열을 만들려면 작성할 것이 많기 때문이다. \LaTeX 패널은 신속한 참조용으로 편리하게 사용할 수 있다.

수학 기호 일부

\LaTeX 에 이용가능한 일부 기호가 다음에 나와 있다.

← `\leftarrow`

 \Leftarrow `\Leftarrow` \Leftrightarrow
$$\geq \quad \$\backslash\geqq\$$$
 \neq `\neq`

∉ \$\\not\\in\$

 ∂ $\$ \backslash \text{partial} \$$ \oint `\oint`

∇ $\$ \backslash nabla \$$

U $\$ \backslash \text{bigcap} \$$

$$\bigcup$$
 \cap $\backslash \text{cap}$ \subset \backslash subset \supseteq \backslash supseteq $\not\supseteq$

⊙ $\$ \backslash \text{bigodot} \$$

\otimes $\$ \backslash \text{bigotimes} \$$

 \oplus `\oplus`

♣ `\clubsuit`

 \perp `\perp` $\vdash \$\vdash$

수천가지 기호를 검색할 수 있는 PDF 파일은 웹사이트를 참조.

www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-

a4.pdf

(Window 메뉴 항목 아래) LaTeX Panel 도 참조한다.

문자 변형

수학 모드에서 텍스트와 기호도 변형할 수 있다.

정규		변형		강조	
$\$R\$$	R	$\$\mathbb{R}\$$	\mathbb{R}	$\$\tilde{R}\$$	\tilde{R}
$\$A\$$	A	$\$\mathcal{A}\$$	\mathcal{A}	$\$\widetilde{A}\$$	\widetilde{A}
$\$x\$$	x	$\$\mathbf{x}\$$	\mathbf{x}	$\$\bar{x}\$$	\bar{x}
$\$p\$$	p	$\$\mathit{p}\$$	p	$\$\hat{p}\$$	\hat{p}
$\$X\$$	X	$\$\mathrm{X}\$$	X	$\$\widehat{X}\$$	\widehat{X}

두가지 다른 강조: $\$\dot{x}\$$ 와 $\$\ddot{x}\$$ 으로 \dot{x} 와 \ddot{x} .

첨자와 지수

윗첨자(예를 들어, x_1)와 아래첨자(예를 들어, 3^2)도 생성할 수 있다:

We can create subscripts (e.g. $x_{\{1\}}$) and superscripts (e.g. $3^{\{2\}}$):

첨자가 단일 문자일 때, 괄호를 생략할 수 있다. 즉, 상기 텍스트에 대해 다음도 동일하게 받아들일 수 있다:

We can create subscripts (e.g. x_1) and superscripts (e.g. 3^2):

만약 윗/아래 첨자 문자가 하나 이상이라면, 괄호를 사용해서 문제를 피한다: 2_{10} 은 2_{10} 을 만들어 낸다. 위첨자와 아래첨자를 동시에 사용할 수 있다: x_{ij}^2 .

분수와 근(root)

$\frac{2+3}{4+5} = \frac{5}{9}$ 처럼 분수 혹은 $\sqrt{81} = 9$ 와 $\sqrt[4]{81} = 3$ 처럼 근(root)도 쉽게 생성할 수 있다.

We can easily create fractions such as $\frac{2+3}{4+5} = \frac{5}{9}$ or roots such as $\sqrt{81}=9$ and $\sqrt[4]{81} = 3$.

물론 분수와 근도 조합할 수 있다: $\frac{\sqrt{4+3}}{\sqrt{16+5}} = \frac{5}{9}$.

And we can combine them as well:

$\frac{\sqrt{4} + 3}{\sqrt{16} + 5} = \frac{5}{9}$.

합과 적분

합과 적분도 생성할 수 있다:

```
\begin{align*}
\sum_{i=0}^{\infty} p^i &= \frac{1}{1-p} & \int_1^2 3x^2 dx &= 7 \\
\sum_{i=0}^{\infty} 0.5^i &= 2 & \int_1^1 3x^2 dx &= 0 \\
\end{align*}
```

결과는 다음과 같다

$$\sum_{i=0}^{\infty} p^i = \frac{1}{1-p} \qquad \int_1^2 3x^2 dx = 7$$

$$\sum_{i=0}^{\infty} 0.5^i = 2 \qquad \int_1^1 3x^2 dx = 0$$

`\nolimits` 와 `\limits` 명령어를 사용해서 \LaTeX 에 디폴트
기본설정된 극한(limit) 표시를 덮어쓸 수 있다.

실습

`eqnarray`* 환경을 사용해서 다음 결과를 생성하시오:

$$\sum_{i=0}^n p^i = \frac{1 - p^{n+1}}{1 - p}$$

예제 일부를 `latex-intro-kr.tex` 파일에서 활용가능하다.

괄호 크기

괄호 크기와 관련한 작은 문제가 왼쪽 식에 나와 있고, 오른쪽에 문제를 고친 식이 나와 있다.

$$\left(\frac{2+3}{4+5}\right) \qquad \left(\frac{2+3}{4+5}\right)$$

상기 표현식에 사용된 코드는 다음과 같다.

```
\begin{align*}
(\frac{2+3}{4+5}) && \left(\frac{2+3}{4+5}\right)
\end{align*}
```

일반적으로 자동으로 크기 조절되는 괄호를 생성하려면, `\left(`, `\left[`, `\left|`, `\left\{` 와 함께 상응하는 오른쪽 괄호를 함께 사용한다. 상기 명령어는 방정식 환경 내부에 있어야만 하고, 왼쪽과 오른쪽 괄호는 항상 짝이 맞아야 된다.

행렬

행렬도 \LaTeX 에서 만들 수 있다:

$$\begin{pmatrix} 4 & 1 & 19 \\ 3 & 8 & 8 \end{pmatrix}$$

상기 행렬을 만드는 코드는 다음과 같다:

```
\begin{eqnarray*}
\left(\begin{array}{ccc} 4 & 1 & 19 \\
3 & 8 & 8 \end{array}\right)
\end{eqnarray*}
```

`array`에 대한 구문은 `tabular` (표)에 사용하는 구문과 같다.

공백과 쌓기(stacking)

`\quad` 을 사용해서 방정식에 공백을 추가할 수 있고, `\stackrel` 을 사용해서 표현식을 쌓을 수도 있다:

```
\begin{eqnarray*}
```

$$E(X+Y) \stackrel{\text{indep.}}{=} E(X) + E(Y)$$

```
\quad\quad
```

$$\text{Var}(X+Y) \stackrel{\text{indep.}}{=} \text{Var}(X) + \text{Var}(Y)$$

```
\end{eqnarray*}
```

상기 명령어는 다음을 만들어 낸다:

$$E(X + Y) \stackrel{\text{indep.}}{=} E(X) + E(Y) \qquad \text{Var}(X + Y) \stackrel{\text{indep.}}{=} \text{Var}(X) + \text{Var}(Y)$$

왜 BibTeX를 사용할까요?

참고문헌을 수작업으로 생성하는 대신에 BibTeX를 사용할 매우 좋은 이유가 상당수 있다.

- 참고문헌 자동 생성.
- 쉬운 참고문헌 스타일 변경.
- 텍스트에서 참조하면서 빼먹은 참고문헌 식별.

BibTeX은 어떻게 동작하나

참고문헌을 만드는데 \LaTeX 과 BibTeX 은 세 단계를 밟는다.

- 인용문을 갖는 문서를 조판할 때 (예를 들어, `\cite{zotova}`), \LaTeX 은 인용문 각각에 대해 메모한다.
- BibTeX 은 작성된 목록을 갖고 출판 데이터베이스에서 각각의 참조를 찾는다.
- 그리고 나서, \LaTeX 에 지시해서 참조하려고 찾은 모든 출판물에 대한 참고문헌을 작성하게 한다.

가장 시간이 많이 소요되는 부분이 최초 데이터베이스를 빌드할 때다. 그 다음부터 동일한 데이터베이스를 반복해서 참조하고, BibTeX은 식은 죽 먹듯 쉽게된다.

BibTeX 자료

- 데이터베이스 생성한다.
- 참조문헌 인용한다.
- BibTeX으로 조판한다.
- 스타일 파일을 빌드한다.

참조 표본 입력

다음과 유사하게 인용하려는 각 항목마다 참조문헌을 생성한다.

```
@article{labelzotova,  
  Author = {Elena Zotova and Charles D Woody and Ehud  
Gruen},  
  Journal = {Brain Research},  
  Pages = {66-78},  
  Title = {Multiple representations ... [etc etc].},  
  Volume = {868},  
  Year = {2000}}
```

참고문헌 구성

각 참고문헌은 출판유형(예를 들어, 책, 기사) 정보가 필요하고, 많은 필드를 포함한다. 예를 들어, 기사(article)입력에 필요한 **필수(required)**필드와 **선택옵션(optional)** 필드가 다음에 나와 있다.

라벨: 참고문서 라벨.

저자(Author) **저널(Journal)** **제목(Title)**

년도(Year) **볼륨(Volume)** **숫자(Number)**

쪽(Pages) **월(Month)** **메모(Note)**

이용가능한 출판 유형 형식목록과 더불어 각 유형에 대해 어느 항목이 필요하고, 어느 항목이 선택옵션인지 확인한다.

http://www.image.ufl.edu/help/latex/entry_bibtex.shtml

수작업에 대한 대안

만약 데이터베이스를 이러한 방식으로 만들고 싶지 않는다면, 다음 서지관리 소프트웨어를 시도해 본다.

- BibDesk: 맥
- JabRef: 맥, 윈도우, 리눅스

BibDesk과 JabRef 서지관리 프로그램 모두 무료이며 온라인에서 이용가능하다. 다른 프로그램도 존재하고 있지만, 맥을 사용하는 저자는 개인적으로 BibDesk만 사용해봤다.

특별한 경우

모호하지 않은 형태로 저자명을 부여하기가 때로는 까다롭다.

- 항상 이름을 {이름(Given Names) 성(Surnames)} 혹은 {성(Surname), 이름(Given Names)} 으로 타이핑한다.
- 괄호안에 감싼 어떤 것이나 단일 항목으로 취급된다 (예를 들어, 저자 = {Maria {San Martino}}).
- 만약 저자가 한명 이상이면, 저자명 각각을 단어 *and*로 분리. *and*가 누군가 이름의 일부라면, 전체 이름을 괄호로 감싼다.
- 강세를 추가할 수도 있다 (즉, $G\{\backslash"o\}_{del}$ 명령어로 Gödel).

다른 많은 뉴앙스가 존재한다. 만약 특별한 이름과 마주치면, 데이터베이스에 넣을 최선의 방법을 온라인 검색한다.

저널 명칭 축약

종종 입력항목에 전체 저널 명칭을 넣고 싶지 않을 때가 있다. 저널 명칭을 축약하려면, *string* 입력 유형을 사용한다:

```
@string{JSS = {Journal of Statistical Software}}
```

축약된 저널 명칭이 사용될 데이터베이스에 이러한 문자열 입력항목이 정의되어야만 된다.

참고문헌 참조

사용될 수 있는 명령어가 네가지가 있다.

- `\cite{라벨이름}` [참조번호], 예를 들어 [1].
- `\citet{라벨이름}` 성 (년도), 예를 들어 Zotova et al. (2000).
- `\citep{라벨이름}` (성, 년도), 예를 들어, (Zotova et al., 2000).
- `\nocite{라벨이름}` 인용되지는 않으나 참고문헌에는 나타남.

첫번째와 마지막은 `uclathes` 클래스에 동작한다. 가운데 두개는 `natbib` 패키지에 사용된다 (학위논문이 아닌 경우 강력히 추천).

본인 문서에 다른 명령어들

참고문헌이 추가되는 장소에 다음 코드 두 줄을 삽입해야만 된다:

```
\bibliographystyle{선택한-스타일}
```

```
\bibliography{데이터베이스-명칭}
```

스타일 명령어는 좀더 위로 이동할 수 있다 (문제가 되지는 않는다). 만약 **natbib** 패키지를 사용한다면, 다른 패키지와 함께 추가한다:

```
\usepackage{natbib}
```

참고문헌 만들기

만약 참고문헌 데이터베이스를 만들고, 인용을 만들고, 텍스트에 참고문헌 참조 명령어를 삽입했다면, 참고문헌을 생성할 준비를 마쳤다. TeXShop에, 마무리하는 간단한 몇단계 절차가 있다:

- \LaTeX 문서를 보통때처럼 조판한다.
- \LaTeX 에서 BibTeX으로 조판 선택옵션을 변경한다:



- BibTeX 으로 다시 조판한다.
- 다시 조판 선택옵션을 \LaTeX 으로 되돌려서 **두번** 이상 컴파일한다.

스타일 파일 빌드하기

BibTeX을 사용하는 큰 장점 중 하나는 참고문헌 스타일과 텍스트 내부 인용을 신속히 변경할 수 있는 역량이다. 이를 위해서 **custom-bib** 프로그램을 사용한다. 다음 웹사이트에서 다운로드한다.

<http://www.ctan.org/tex-archive/help/Catalogue/entries/custom-bib.html>

custom-bib 패키지는 **latexTemp** zip 압축파일에 포함되어 있다.

스타일 파일 빌드하기

latexTemp > custom-bib 디렉토리를 열고, **makebst.tex** 파일을 연다. 프로그램을 실행시키기 위해서,

- (1) 파일을 열어서 조판한다.
- (2) 추가 디렉토를 얻어오는 첫번째 질문에 YES를 타이핑한다.
- (3) 적절한 파일 명칭을 선택한다 (확장자를 추가할 필요는 없다).
- (4) 각 스타일 질문에 대답한다.
- (5) 마지막 질문에, *Finished!! ... Shall I now run this batch job?* (NO), YES 를 타이핑한다.

(3) 단계에서 명명한 .bst 확장자를 갖는 파일을 찾아 복사한다. 해당 스타일을 적용할 참고문헌을 생성할 파일을 갖는 폴더에 두거나 혹은 참고문헌 폴더에 둔다 (L^AT_EX 문서에 스타일 파일로 이것을 참조한다).

실습

`latex-intro-kr.tex` 파일을 열고 마지막 절로 간다.

`\citet{victor}`의 참고문헌을 추가한다. 또한 `\citep{victor}` 로
참조대상을 추가하고 조판한다 (4단계 모든 단계). 참고문헌
사이에 차이는 무엇인가요? 논문에서 각각을 어떻게 사용할까요?

사용자 정의 명령어 재료

- 계수기(Counters)
- 명령어 생성하기
- 환경 생성하기

기존 계수기

\LaTeX 은 계수기(변수)를 사용해서 적절하게 번호를 매긴다.

- \LaTeX 에서 사용되는 계수가 다음에 나와있다: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `page`, `footnote`, `equation`, `figure`, `table`. 이러한 계수기는 상응하는 명령어에 대응된다.
- 각 수준별 열거할 때 다른 계수기가 사용된다: `enumi`, `enumii`, `enumiii`, `enumiv`.
- 다른 \LaTeX 계수기 몇가지: `paragraph`, `subparagraph`, `mpfootnote`.

신규 계수기 생성하기

본인 목적에 맞춰서 자신만의 계수기를 생성할 수도 있다. 아마도 번호 매기고 싶은 예제가 있을 수 있다.

```
\newcounter{counterName}[inCounter]
```

상기 명령어는 counterName 으로 불리는 새로운 계수기를 생성한다. inCounter 인자와 + 꺾쇠 괄호는 선택옵션이다. inCounter가 증가할 때마다 inCounter 를 사용해서 counterName 을 재설정한다. (예를 들어, subsection 은 “inCounter” section 을 갖고 있다).

변형하기

기존 혹은 신규 계수기를 변형할 수 있다.

```
\setcounter{counter}{n}
```

```
\addtocounter{counter}{n}
```

```
\stepcounter{counter}
```

```
\refstepcounter{counter}
```

만약 `\label` 을 붙여놓는다면, `\refstepcounter` 명령어를 사용해서 계수기 값을 참조한다.

출력하기

계수기를 생성하고, 변형하고, 참조할 수 있다. 하지만, 문서에서 계수기를 출력할 필요도 있다. 계수기를 다음 명령어 중 하나를 호출해서 출력한다:

`\arabic{chapter}` (4, 아라비아 숫자)

`\Roman` (IV, 대문자 로마 숫자)

`\roman` (iv, 소문자 로마 숫자)

`\Alph` (D, 영문 대문자)

`\alph` (d, 영문 소문자)

`\fnsymbol` (§, 각주 기호)

사용자 정의 명령어와 환경에 계수기를 두고 사용한다.

간단한 명령어

x_1, \dots, x_n 같은 흔한 문장은 다음 new 명령어로 축약할 수 있다.

```
\newcommand{\xvec}{x_1,\dots,x_n}
```

상기 명령어를 삽입하고 나서 (나중에 문서에) $\$ \xvec \$$ 을 타이핑하면, x_1, \dots, x_n 을 얻게 된다. 만약 달러 부호를 생략하면 문제가 된다. 이것을 해결하는데 추가 명령어를 사용한다.

```
\newcommand{\xvec}{\ensuremath{x_1,\dots,x_n} }
```

두번째 정의 말미에 추가 공백을 두어서, 공백 문제를 방지하게 했다. 좀더 멋진 해결책은 **xspace** 패키지를 사용하는 것이다 (*Guide to LaTeX*, 186쪽을 참조한다).

인자를 갖는 명령어

만약 작성한 명령어를 일반화하려고 하면, 인자 두개를 추가한다.

```
\newcommand{\subvec}[2]{\ensuremath{\#1_1,\dots,\#1_{\#2}}}
```

`\subvec{y}{m}`로부터 y_1, \dots, y_m 을 생성할 수 있다.

n^{th} 인자에 대해서 `\#n` 을 통해 추가 인자를 생성하고 참조할 수 있다. 디폴트 선택옵션 인자도 활용할 수 있다 (*Guide to LaTeX*, 188쪽을 참조한다).

일반화

신규 명령어에 대한 일반적인 프레임워크는 다음과 같다.

```
\newcommand{\commandName}[n]{the commands}
```

여기서

- `commandName`은 명령어 명칭이 된다.
- `n`은 인자 갯수가 된다.
- 인자를 `the commands`에 `#1`, `#2`, ..., `#n`으로 참조한다.

이미 존재하는 명령어를 다시 정의하려면, 상기와 동일한 형식으로 `\renewcommand` 명령어를 사용한다.

견본 환경

환경은 `begin`과 `end` 태그를 사용한다 (예를 들면, `itemize`). `\begin`과 `\end` 태그에 발생하는 것만 정의할 필요가 있다. 예를 들어,

```
\newenvironment{예제}
```

```
{\small\textbf{예제.} \hspace{2mm}} % 시작하는 것
```

```
{\\} % 끝나는 것
```

견본 환경 호출:

```
\begin{예제}
```

나머지 연산의 덧셈은 신비롭게 동작한다: $2+2=1 \pmod{3}$.

```
\end{예제}
```

결과:

예제. 나머지 연산의 덧셈은 신비롭게 동작한다: $2 + 2 = 1 \pmod{3}$.

일반 환경

일반적으로 환경(environment)은 다음과 같은 형태를 갖는다.

```
\newenvironment{environmentName}{begin stuff}{end stuff}
```

또한, n 개 인자를 선언할 수도 있다.

```
\newenvironment{environmentName}[n]{begin stuff}{end  
stuff}
```

앞에서처럼, 시작하는 것(begin) and 끝나는 것(end) 에서 인자를 $\#1, \dots, \#n$ 으로 참조한다.

기존 환경을 다시 정의하려면, 위와 동일한 형식으로

`\renewenvironment` 명령어를 사용한다.

환경 + 계수기

```
\newcounter{example}
```

```
\setcounter{example}{0}
```

```
\newenvironment{example}
```

```
{\refstepcounter{example}\small
```

```
\textbf{Example \arabic{example}.}\hspace{2mm}}
```

```
{\\}
```

체계적 정리와 시간 절약

`\include` 명령어는 장문 문서에 유용하다:

```
\include{otherDocName}
```

예를 들어, 본 발표자료는 실제로 개별 문서 세개를 호출한다: 각각 대단원마다 하나씩. 그렇게 함으로써 문서를 체계적으로 관리하면서도 작업하지 않는 문서 부분을 조판하지 않음으로써 시간을 절약할 수 있다:

```
\include{math/math} % “math” 문서는 “math” 폴더에 있다
%\include{bibtex/bibtex}
%\include{comenv/comenv}
```

요약정리

이번 학습을 통해서 \LaTeX 다음 기능에 대한 전반적인 이해를 하게 되었다.

- \LaTeX 에서 수학 모드 사용
- BibTeX을 사용해서 참고문헌 생성
- 사용자 정의 명령어와 환경 생성

질문이 있으십니까?