

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

Chapter 3~4  
수식 조판하기  
특별한 기능

KINS

2012년 12월 31일

## 제 3장

### 수식 조판하기

## 준비 사항

수식을 조판하기 전에 꼭 사용해야 하는 패키지가 있다. 바로  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 의 패키지 형태인 `amsmath`이다.

수식 조판을 할 일이 있다면 항상 이 패키지를 로드해놓는 습관을 들이자.

```
\usepackage{amsmath}
```

# 기본 사항

수식은 두 가지 형태로 식자할 수 있는데, 본문 중간에서 쓰이는 inline 형식과 별도 단락을 사용하는 display 형식이 있다.

- ▶  $\$$   $\$$  사이에 들어가는 모든 것은 inline 수식으로 취급된다.
- ▶  $\backslash[ \backslash]$  사이에 들어가는 모든 것은 display 수식으로 취급된다.

inline 수식과 display 수식은 식자하는 방식이 조금 다르다.

# 기본 사항

우리는  $\sum_{k=1}^n k = \frac{n(n+1)}{2}$  임을 알고 있다.

우리는  $\sum_{k=1}^n k = \frac{n(n+1)}{2}$  임을 알고 있다.

우리는  $\sum_{k=1}^n k = \frac{n(n+1)}{2}$  임을 알고 있다.

우리는

$$\sum_{k=1}^n k = \frac{n(n+1)}{2}$$

임을 알고 있다.

## 수식모드와 텍스트모드의 차이

- ▶ 수식 모드 안에서 모든 텍스트는 '변수'로 처리되어 *italic* 폰트로 식자된다. 곧은 글자를 쓰고 싶다면  
`\mathrm{}`, `\textrm{}`  
등의 명령어를 써야 한다.
- ▶ 입력파일의 모든 여백은 무시된다. 여백을 주기 위해서는 특수한 명령어  
`~`, `\`, `,`, `\;`, `\:`, `\quad`, `\qquad`  
등을 사용해야 한다.
- ▶ 입력파일의 빈 줄은 허용되지 않는다. 즉, 수식을 여러 문단에 걸쳐 적을 수 없다.

# 수식 번호 넣기

단일 display수식에 번호를 넣으려면

```
\begin{equation}
```

수식

```
\end{equation}
```

환경을 사용한다. 수식 번호는 문서 클래스에 따라 수식 우측에 자동으로 붙여진다.

equation은 번호를 붙이는 환경이므로 `\label{}`명령과 `\ref{}`명령 또는 `\eqref{}`을 이용해서 참조할 수 있다.

## 수식 번호 넣기

우리는

```
\begin{equation}\label{sum}
\sum_{k=1}^n k
=\frac{n(n+1)}{2}
\end{equation}
임을 알고 있다. 식 \ref{sum}
에서\ldots
```

우리는

$$\sum_{k=1}^n k = \frac{n(n+1)}{2} \quad (1)$$

임을 알고 있다. 식 ?? 에  
서...

## eqnarray환경

여러 줄에 걸친 수식을 입력하고자 한다면 eqnarray환경을 써야 한다. *이 환경은 rcl형태의 표처럼 동작한다.*

```
\begin{eqnarray}
수식 & 수식 & 수식 \\
수식 & 수식 & 수식 \\
...
\end{eqnarray}
```

eqnarray환경에서는 각 줄에 있는 수식마다 번호를 자동으로 붙인다. 번호 붙이기를 원하지 않으면 `\begin{eqnarray*}`환경을 쓰거나 번호가 붙지 않길 원하는 행의 `\\` 바로 앞에 `\nonumber`명령을 쓰면 된다.

## eqnarray 환경

다음은 자명하다:

```
\begin{eqnarray}
F &= & ma \\
E &= & mc^2 \\
F &= & \frac{GMm}{r^2}.
\end{eqnarray}
```

아주 쉽다.

다음은 자명하다:

$$F = ma \quad (2)$$

$$E = mc^2$$

$$F = \frac{GMm}{r^2}. \quad (3)$$

아주 쉽다.

## align환경

amsmath패키지에 정의된 align환경은 여러 줄에 걸친 수식을 하나의 기준으로 정렬해 식자해준다. 행의 끝은 \\으로 나타내며, 정렬 기준 (열 나눔)은 &으로 한다. **한 행에 &는 꼭 한 번만 와야 한다.**

```
\begin{align}
```

```
x^2+2x+1&=0\\
```

$$x^2 + 2x + 1 = 0 \quad (4)$$

```
(x+1)^2&=0\\
```

$$(x + 1)^2 = 0 \quad (5)$$

```
x&=-1
```

$$x = -1 \quad (6)$$

```
\end{align}
```

\nonumber과 별표 붙은 환경 align\*은 eqnarray환경에서와 같은 방식으로 작동한다.

## gather환경

gather 환경은 amsmath 패키지에 정의되어 있으며 여러 줄에 걸친 수식을 가운데 정렬해주는 환경이다. 각 행은 \\로 구분하며, **열을 구분하는 인자는 없다.**

```
\begin{gather}
```

$$1 \\ \qquad \qquad \qquad 1 \qquad \qquad \qquad (7)$$

$$x+y \\ \qquad \qquad \qquad x + y \qquad \qquad \qquad (8)$$

$$x^2+2xy+y^2 \\ \qquad \qquad \qquad x^2 + 2xy + y^2 \qquad \qquad \qquad (9)$$

$$x^3+3x^2y+3xy^2+y^3 \\ \qquad \qquad \qquad x^3 + 3x^2y + 3xy^2 + y^3 \qquad \qquad \qquad (10)$$

```
\end{gather}
```

\nonumber명령어와 별표 붙은 환경 gather\*역시 같은 방법으로 작동한다.

## multiline환경

amsmath패키지에 정의된 multiline환경은 한 줄에 쓸 수 없는 긴 수식을 여러 줄에 걸쳐 식자할 때 유용하다. 수식의 행 나눔은 \\ 명령어를 사용하며 **열 나눔은 없다**. 이 때, 첫 행은 좌측정렬, 마지막 행은 우측정렬되고 나머지는 가운데정렬된다. 수식번호는 자동으로 마지막행에만 붙는다.

```
\begin{multiline}
(v+w+x+y+z)^2=v^2+w^2+x^2+y^2+z^2\\
+2vw+2vx+2vy+2vz+2wx+2wy\\+2wz+2xy+2xz+2yz
\end{multiline}
```

$$\begin{aligned}(v + w + x + y + z)^2 &= v^2 + w^2 + x^2 + y^2 + z^2 \\ &+ 2vw + 2vx + 2vy + 2vz + 2wx + 2wy \\ &+ 2wz + 2xy + 2xz + 2yz \quad (11)\end{aligned}$$

## Remark

- ▶ 문서클래스의 option 중, leqno는 수식번호를 수식 왼쪽에 식자한다.
- ▶ fleqn옵션을 사용하면 display형태의 수식이 왼쪽 정렬된다.
- ▶ 긴 수식을 정렬할 때, 수식을 어느 시점에서 다음 줄로 넘겨야 할지는 사람이 결정해야 한다.

## 주요 명령어

- ▶ 위첨자와 아래첨자는 `^`와 `_`를 사용하여 나타낸다. 이 때, 한 문자 이상이 첨자로 들어가게 하려면 중괄호 `{}`로 묶어주어야 한다. 또한, 첨자 자리에 여러 줄에 걸친 수식을 쓰려면 `amsmath` 패키지의 `\substack{}`명령어를 쓴다. `substack`안에서 줄바꿈은 `//`명령어를 사용한다.
- ▶ `\overbrace{}`<sup>`^`</sup>와 `\underbrace{}`<sub>`_`</sub>는 각각 수식의 위,아래에 수평 중괄호를 입력해준다.
- ▶ 이항계수는 `amsmath`패키지의 `\binom{ }{ }`명령을 사용한다.

## 주요 명령어

- ▶ mod기호는 amsmath패키지의 `\pmod{}`혹은 `\mod{}`명령어를 쓴다.
- ▶ `cfrac{ }{ }`은 연분수를 쓸 때 사용하면 된다.
- ▶ 각종 함수의 이름은 변수와 구분되도록 곧은 글꼴(roman)로 표기해야 한다. ( $\sin\theta(x)$ ,  $\sin\theta(o)$ )  
사용자의 편의를 위해  $\text{\LaTeX}$ 은 몇몇 함수를 미리 명령어로 지정해놓았는데, lshort 53페이지에 미리 정의된 함수의 목록이 있다.

## 예시

```
\begin{eqnarray*}
n&=&\underbrace{1+1+\cdots+1}_{n\text{번}}\\
\binom{n}{m}&=&\frac{n!}{m!(n-m)!}\\
T^{\mu}_{\nu}&=&g_{\nu\eta}\rho U^{\mu}U^{\eta}\\
c^2&=&a^2+b^2-2ab\cos\theta
\end{eqnarray*}
```

$$n = \underbrace{1 + 1 + \cdots + 1}_n$$

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

$$T_{\nu}^{\mu} = g_{\nu\eta}\rho U^{\mu}U^{\eta}$$

$$c^2 = a^2 + b^2 - 2ab\cos\theta$$

## 예시

```
\begin{align*}x&=\cfrac{1}{1+\cfrac{1}{1+\cfrac{1}{1+\ddots}}}\backslash \\ 1&\equiv 10 \pmod{3}\backslash \\ \frac{\pi^2}{3}&= \\ \sum_{\substack{n \in \mathbb{Z} \\ n \neq 0}} \frac{1}{n^2} \\ \end{align*}
```

$$x = \frac{1}{1 + \frac{1}{1 + \frac{1}{1 + \ddots}}}$$

$$1 \equiv 10 \pmod{3}$$

$$\frac{\pi^2}{3} = \sum_{\substack{n \in \mathbb{Z} \\ n \neq 0}} \frac{1}{n^2}$$

## 짜맞춤

수학 모드에서 짜맞춤(괄호로 묶기)는 가독성을 높이는 중요한 요소이다.

$$1 + \left(\frac{1}{1+x^2}\right)^2 \quad \text{vs.} \quad 1 + \left(\frac{1}{1+x^2}\right)^2$$

이렇게 적절한 짜맞춤은 `\left...`와 `\right...`를 사용하여 할 수 있다.

이 때, `\left`를 사용했다면 반드시 `\right`로 닫아주어야 한다.

오른쪽 짜맞춤 요소를 식자하기 싫은 경우 논리적 짜맞춤기호인 `\right.`를 쓴다.

## 짜맞춤-예시

```
\begin{eqnarray*}
\left| \frac{1}{2}\psi \right\rangle &= & \frac{1}{2} \left| \psi \right\rangle \\
\left\{ 1 - \left( \frac{x}{x-1} - 1 \right) \right\}^2 & & \\
&= & \left( \frac{x-2}{x-1} \right)^2 \\
\end{eqnarray*}
```

$$\left| \frac{1}{2}\psi \right\rangle = \frac{1}{2} |\psi\rangle$$

$$\left\{ 1 - \left( \frac{x}{x-1} - 1 \right) \right\}^2 = \left( \frac{x-2}{x-1} \right)^2$$

## 수학 글꼴

수학 모드에서만 사용할 수 있는 글꼴이 있다. 이들 중 대부분은 `amssymb` 또는, `amsfonts`패키지를 필요로 하며 `\mathscr{}`의 경우 `mathrsfs`패키지를 필요로 한다.

보기	명령어	명칭
ABCDEabcde12345	<code>\mathrm{...}</code>	Roman
<b>ABCDEabcde12345</b>	<code>\mathbf{...}</code>	Boldface
ABCDE∂∫∫∫∫∫	<code>\mathbb{...}</code>	Blockboard bold
<i>ABCDE</i> ∪ ∩ ∞ ∈ ∃ Δ ∇	<code>\mathcal{...}</code>	Calligraphic
<i>A B C D E</i>	<code>\mathscr{...}</code>	Script
<b>A B C D E</b> abcde12345	<code>\mathfrak{...}</code>	Fraktur

이제 기본적인 수식을 입력할 준비는 모두 끝났다. 남은 문제는 원하는 기호의  $\text{\LaTeX}$ 코드를 모르는 경우가 많다는 것인데, 다음 두 곳이 도움이 된다.

- ▶ lshort 63페이지에서 69페이지에는 거의 모든 수학 연산과 꾸밈기호, 짝맞춤기호가 나와 있다.
- ▶ 위에서 찾지 못한 것은 <http://detexify.kirelabs.org/>에서 찾을 수 있을 것이다.

▶ 연습문제 1. 다음을 식자하시오.

$$i\hbar \frac{d}{dt} |\Psi(t)\rangle = \mathcal{H}(t) |\Psi(t)\rangle \quad (1)$$

$$R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = 8\pi GT_{\mu\nu}$$
$$\oint \vec{E} \cdot d\vec{A} = \frac{Q_{\text{enc}}}{\epsilon_0} \quad (2)$$

$$\nabla \times \mathbf{B} = \frac{1}{c} \left( 4\pi \mathbf{J} + \frac{\partial \mathbf{E}}{\partial t} \right) \quad (3)$$

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} \quad (4)$$

$$\wp(z) = \frac{1}{z^2} + \sum_{\omega \in \Lambda^*} \left( \frac{1}{(z + \omega)^2} - \frac{1}{\omega^2} \right) \quad (5)$$

## 행렬과 조건식 조판

이제 array형태의 수식 입력에 대해 알아보자. 이것을 배우면 행렬과 조건식을 식자할 수 있다. array형태의 수식을 입력하려면 수학 모드에서 array 환경을 사용한다.

```
\[
\begin{array}{ array format }
.....&.....&.....&.....\\
&...&...&...\\
...
\end{array}
\]
```

array환경의 작동방식은 tabular와 아주 유사하다. array format의 자리엔 tabular와 마찬가지로 r, l, c와 열 구분자가 들어간다.

## 행렬과 조건식 조판 - 예시

```
\[  
\mathbf{M}=  
\left[  
\begin{array}{cc}  
a&b\\  
c&d  
\end{array}  
\right]  
\]
```

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

## 행렬과 조건식 조판

조건이 붙은 식을 조판하는 것 역시 비슷한 원리이다.

```
\[
```

```
\chi_A (x)=
```

```
\left\{
```

```
\begin{array}{ccc}
```

```
1& \text{trm}{if} & x\in A \ \ \
```

```
0& \text{trm}{if} & x\notin A
```

```
\end{array}
```

```
\right.
```

```
\]
```

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

이 때, 짝맞춤을 위해 오른쪽을 `\right.`로 꼭 닫아야 한다.

## cases환경

앞에서 한 것 처럼 복잡하게 조건식을 조판할 수도 있지만, amsmath 패키지에 있는 cases환경을 쓰면 대단히 편리하게 할 수도 있다. cases 환경은 **수식 환경 안에서만 작동**하며, 다음과 같이 쓴다.

```
\[
\chi_A (x)=
\begin{cases}
1 & \text{if } x \in A \\
0 & \text{if } x \notin A
\end{cases}
\]
```

$$\chi_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

이 때, cases환경 내의 `\text{}`명령어 안에서 수식을 쓰려면 수식 환경을 안에 써야 한다.

## matrix환경

행렬 조판 역시 amsmath패키지의 pmatrix, bmatrix, vmatrix환경을 쓰면 편리하다. **이들 환경은 array와 동작 방식이 같지만 format 인자를 받지 않으며** 입력에 따라 자동으로 크기가 조절된다.

```
\begin{align*}
A&=
\begin{pmatrix}
a & b & c \\
d & e & f
\end{pmatrix} \\
B &= \begin{bmatrix}
2 & 3 \\
5 & 7
\end{bmatrix}
\end{align*}
```

$$A = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}$$

$$B = \begin{bmatrix} 2 & 3 \\ 5 & 7 \end{bmatrix}$$

- ▶ 연습문제 2. 다음을 조판하시오.

$$\mathbf{M} = \left[ \begin{array}{c|ccc} 1 & 0 & 0 & 0 \\ \hline 0 & a & b & 0 \\ 0 & c & d & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right]$$

[**HINT.** `\multicolumn{}{}{} 명령어를 활용한다.`]

## 정리와 법칙

정리, Lemma, 정의 또는 이와 유사한 (numbering 이 필요한) 요소를 조판하는 방법을 알아보자.

$\LaTeX$ 은 이러한 요소를 사용자가 정의한 numbered environment로 이해한다. 즉 각 요소에 대한 새로운 환경을 다음과 같이 만들어주어야 한다.

## 정리와 법칙

새로운 정리와 법칙 환경을 선언하는 방법은 다음과 같이 두 가지이다.

▶ **Type 1.**

```
\newtheorem{ name }{ text }[ reset ]
```

▶ **Type 2.**

```
\newtheorem{ name }[ counter ]{ text }
```

이렇게 선언된 정리와 법칙 환경을 본문에서 사용하려면 다음과 같이 한다.

```
\begin{ name }[ description ]  
...contents...  
\end{ name }
```

## 정리와 법칙

- ▶ **Type 1.** `\newtheorem{ name }{ text }[ reset ]`
- ▶ **Type 2.** `\newtheorem{ name }[ counter ]{ text }`

인자	설명
name	새로 정의될 환경이 가질 $\LaTeX$ 입력 파일 상에서의 <b>호출이름</b> 이다.
text	환경이 본문에서 쓰였을 때, 실제로 문서에 찍힐 <b>말머리</b> 이다.
reset	환경의 카운터가 의존할 <b>이미 정의된</b> 카운터를 지정한다. reset에 해당하는 카운터가 바뀔 때 마다 해당 환경 카운터가 초기화된다.
counter	환경이 공유할, <b>이미 정의된</b> 카운터를 지정한다.

## 정리와 법칙

- ▶ **Type 1.** 방식으로 선언하면,  $\LaTeX$ 은 새 환경을 정의함과 동시에 `name`을 이름으로 하는 새로운 카운터를 만들고 이 카운터를 `reset`이라는 카운터에 종속시킨다.
- ▶ **Type 2.** 방식으로 선언하면  $\LaTeX$ 은 새 환경을 정의하지만 카운터는 생성하지 않으며, 해당 환경의 `numbering`은 `counter` 인자가 지정한 카운터를 그대로 이어받는다.

# 정리와 법칙 -예시

```
\newtheorem{thm}{Theorem}[section]
\newtheorem{cor}[thm]{Corollary}
\newtheorem{hhh}{Definition}[section]
\begin{hhh}
생각할 수 있는 동물을 사람이라 한다.
\end{hhh}
\begin{thm}
모든 사람은 죽는다.
\end{thm}
\begin{cor}[소크라테스]
소크라테스는 죽는다.
\end{cor}
```

## Definition

생각할 수 있는 동물을  
사람이라 한다.

## Theorem

모든 사람은 죽는다.

## Corollary (소크라테스)

소크라테스는 죽는다.

## 제 4장

### 특별한 기능

# 참고문헌 만들기

참고문헌 목록을 만들기 위해서는 thebibliography 환경을 쓴다.

```
\begin{thebibliography}{ n }  
\bibitem[ label ]{ marker } .....  
\bibitem[ label ]{ marker } .....  
...  
\end{thebibliography}
```

- ▶ 인자 n은 참고문헌의 갯수가 n개 이하임 명시하는 것이다.
- ▶ 참고문헌을 본문에서 인용하려면 \cite{ marker } 명령을 사용한다.

## 참고문헌 만들기 - 예시

박승쿵은 2011년 논문 `\cite{park}` 에서 골드바흐의 추측을 증명하였다.

```
\begin{thebibliography}{5}
\bibitem{park} Park. S,
\emph{Ann. of Math.,}
\textbf{999}, 99, (2011)
\end{thebibliography}
```

박승쿵은 2011년 논문 [?] 에서 골드바흐의 추측을 증명하였다.



Park. S, **Ann. of Math.**, **999**, 99, (2011)

# BibTeX의 사용

다음과 같은 절차로 참고문헌목록을 쉽게 만들 수 있다.

1. .bib (bibliography database)파일을 작성한다. 대부분의 저널은 각 논문에 대해 bibtex으로 export할 수 있게 되어 있다.
2. .bst (bibliography style)파일을 작성한다. 이것 역시 각 저널마다 만들어 놓은 양식이 있다.
3. .tex 문서를 작성하면서 인용을 원하는 곳에 `\cite{}`명령을 이용해 인용한다.
4. .tex 문서에서 참고문헌목록이 들어갈 곳에 `\bibliography{.bib 파일 이름}`과 `\bibliographystyle{.bst 파일 이름}`을 적는다.
5. pdfL<sup>A</sup>T<sub>E</sub>X → BibT<sub>E</sub>X → pdfL<sup>A</sup>T<sub>E</sub>X → pdfL<sup>A</sup>T<sub>E</sub>X순으로 컴파일한다.

## 하이퍼텍스트링크와 책갈피

문서 내에서 상호참조한 부분이나 외부 링크를 인용한 곳에 하이퍼링크를 만들어주는 도구가 있다. 이를 이용하려면,

```
\usepackage{dhucs-ucshyper}
```

패키지를 사용한다. 이 패키지를 로드하면 모든 상호참조에 자동으로 하이퍼링크가 생성되고 .pdf출력물에 책갈피가 자동 생성된다.

## 하이퍼텍스트링크와 책갈피

또한, 추가적으로 `\href{ link }{ appearance }` 명령을 쓸 수 있다.

```
\href{http://www.kaist.ac.kr}{눌러보세요}
```

눌러보세요

# Beamer

Beamer 클래스를 사용하면  $\LaTeX$ 으로 발표용 슬라이드를 만들 수 있다.

```
\documentclass[ option ]{beamer}
```

# Beamer

Beamer 클래스를 사용하면  $\text{\LaTeX}$ 으로 발표용 슬라이드를 만들 수 있다.

```
\documentclass[ option ]{beamer}
```

하지만 아주 특별한 경우가 아니라면 ppt를 쓰는 편이 좋다.

- ▶ 숙제 2. 다음 문서를 조판하시오.