

koredumath.sty
중·고등학교 교육용 수학식 표현

Nova de Hi

2025-05-14 version 0.5

차례

1 이 문서를 읽는 방법	1
2 패키지 옵션과 의존성	1
3 명령과 환경	1
3.1 폰트	1
3.2 분수	2
3.3 근호	2
3.4 벡터	2
3.5 수열, 미분, 적분	3
3.6 합동, 닮음, 평행	6
3.7 등호 부등호	6
3.8 간격, 좌표, 괄호	6
3.9 집합과 논리	7
3.10 cases	8
3.11 행렬	8
3.12 그밖의 기호	9
4 실험적 기능	10
4.1 첨자의 위치	10
4.2 equation의 번호	11
5 마치는 말	14
감사의 말	14
찾아보기	16

1 이 문서를 읽는 방법

koredumath는 jpnedumathsymbols를 바탕으로 이를 수정하거나 추가·보충하는 방식으로 제작하였다. 그러므로 많은 명령과 환경이 jpnedumathsymbols에 정의된 것을 쓴다. 이를 위해서는 해당 패키지의 패키지 문서를 읽어야 할 것이지만 사용자 매뉴얼이라는 관점에서 번거로움을 피할 수 없으므로 편의를 위하여 jpnedumathsymbols의 명령도 소개하고 예제를 그대로 가져다 쓰기로 한다. 명령이나 환경의 끝에 (J)가 붙어 있는 것은 jpnedumathsymbols의 것을 그대로 사용한다는 의미이다.

명령어를 표기한 곳에서 백슬래시가 붙어 있으면 명령이고 없으면 환경의 이름이다.

2 패키지 옵션과 의존성

[`similarsignfig`] ∞ 를 그림으로 식자한다. default는 false.

[`similarsignfont`]= ∞ 부호를 식자할 폰트를 지정한다.

[`eqnformat`]=<format> circ, dots, circdots, dotscirc, none 가운데 하나를 선택한다.

[`noamssymb`] 만약 unicode-math를 쓰려 하는 경우라면 amssymb를 로드하지 않는 것이 좋다. jpnedumathsymbols는 amssymb를 자동으로 로딩하는데 이것을 억제한다. 단 이 패키지를 \usepackage하기 전에 명시적으로 amssymb를 로드한 경우에는 효력이 없다.

그밖의 옵션 다른 모든 옵션은 jpnedumathsymbols 패키지에 전달한다. 예컨대 [`mathbb`] 등.

이 패키지는 T_EXLive의 jpnedumathsymbols 이외에, [KTUG Private Repository](#)를 통하여 배포하는 kocircnum을 요구한다.

3 명령과 환경

3.1 폰트

`\AA, \BB, ... (J)` `\mathrm{A}`, `\mathrm{B}`와 같이 math roman체 대문자를 표현한다.

점 `\AA`, 선분 `\overline{\AA\BB}`, 삼각형 `\AA\BB\CC`
 정규분포 `\NN(m, \sigma^2)`, 확률 `\PP(\abs{Z} \leq 1.96) = 0.95`

점 A, 선분 $\overline{AB}$, 삼각형 ABC
 정규분포 $N(m, \sigma^2)$, 확률 $P(|Z| \leq 1.96) = 0.95$

`\Z, \N, \Q, \NZ, \NP, \R, \C (J)`

수집합을 나타내는 명령이다. [`mathbb`] 옵션이 주어져 있으면 `\mathbb`로, 그렇지 않으면 `\mathbf`로 식자한다.

`\[\N\NZ\NP\Z\Q\R\C]`

$\mathbb{N}_0\mathbb{N}_+\mathbb{Z}\mathbb{Q}\mathbb{R}\mathbb{C}$

실제 중·고등학교 수학적 표현에서 이러한 집합 기호를 거의 쓰지 않는 것으로 안다. 예컨대 $x \in \mathbb{R}$ 과 같은 표현 대신 “ x 는 실수”라고 쓰도록 하고 있다. 또한 $\mathbb{R}$ 나 $\mathbf{R}$ 와 같은 수학 폰트는 사용하지 않는다. 비근한 예로, 기댓값을 나타내는 $\mathbb{E}(X)$ 와 같은 표현을 단순히 $E(X)$ 로 쓰도록 하고 있다. 그러나 중·고등과정을 넘어서면 이러한 기호가 필요해진다.

3.2 분수

$\frac{}{} (J)$ $\frac{}{} (J)$ 명령으로 만들어지는 분수는 인라인 수식이라도 큰 분수를 식자한다. $\originalfrac$ 은 $\amsmath$ 의 $\frac{}{} (J)$ 이다. 다만 지수나 첨자 위치에 들어갈 때에는 작은 분수가 된다.

$\frac{2}{5+3}$, $\frac{\frac{a}{b}}{c}$, $\frac{\frac{a}{b}}{c}$, $a^{\frac{2}{3}}$

$$\frac{2}{5+3}, \frac{\frac{a}{b}}{c}, \frac{\frac{a}{b}}{c}, a^{\frac{2}{3}}$$

3.3 근호

$\sqrt{} (J)$ 근호 명령은 $\jpnedumathsymbols$ 에 일본식 근호로 수정되어 있다. $\koredumath$ 는 이를 채택하지 않고 $\amsmath$ 의 $\sqrt{} (J)$ 를 그대로 사용하게 하였다. 일본식 근호가 필요하다면 $\jpnsqrt$ 를 쓴다.

$\sqrt{2}$, $\sqrt[3]{2}$, $\sqrt[3]{2}$, $\sqrt[3]{2}$

$$\sqrt{2}, \sqrt[3]{2}, \sqrt[3]{2}, \sqrt[3]{2}$$

3.4 벡터

$\vec{} (J)$ 문자 위에 화살표로 벡터를 표시할 때 사용한다. $\LaTeX$ 표준 $\vec{} (J)$ 명령에 비하여 화살표의 위치와 길이가 조절되어 있다.

$\vec{a}$, $\vec{b}$, $\vec{x}$, $\vec{y}$, $\vec{AA\ BB}$

$$\vec{a}, \vec{b}, \vec{x}, \vec{y}, \vec{AB}$$

$\rvec, \rvec* (J)$ 별표가 붙지 않은 것은 2차원 행벡터, 별표가 붙으면 3차원 행벡터이다. 각 원소를 모두 하나씩 인자로 준다.

$\rvec{1}{2}$, $\rvec*{1}{2}{3}$

$$(1, 2), (1, 2, 3)$$

패키지 옵션 `[rvecbracket]`이 지정되어 있으면 괄호(parenthesis) 대신 대괄호(bracket)로 표현한다.

$\cvec, \cvec* (J)$ 별표가 붙지 않은 것은 2차원 열벡터, 별표가 붙으면 3차원 열벡터이다. 각 원소를 모두 하나씩 인자로 준다.

`\cvec{1}{2}`, `\cvec*{1}{2}{3}`

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

패키지 옵션 `[cvecbracket]`이 지정되어 있으면 괄호 대신 대괄호로 표현한다.

참고 행렬과 마찬가지로 열벡터의 행간격은 현재 문서의 `\linespread`에 영향을 받는다.

`\inp`, `\innerproduct` (J)

벡터의 내적을 표현한다. 기본값은 점으로 나타내는 것이다.

`\innerproduct{\vec{a}}{\frac{\vec{b}}{2}}`
 $\hookrightarrow$ `=\inp{\vec{a}}{\frac{\vec{b}}{2}}`,
 $\hookrightarrow$ `\innerproduct*{\vec{a}}{\frac{\vec{b}}{2}}`

$$\vec{a} \cdot \frac{\vec{b}}{2} = \vec{a} \cdot \frac{\vec{b}}{2}, \vec{a} \cdot \frac{\vec{b}}{2}$$

만약 `[innerproductbracket]` 옵션이 주어지면, 점 대신 `bracket`으로 표현한다.

`\inpbracket`

이 기호를 별도의 옵션 없이도 얻을 수 있도록 하기 위해 `\inpbracket`을 따로 정의해 두었다.

`\[\inpbracket{\vec{a}}{\frac{\vec{b}}{2}}\]`

$$\left\langle \vec{a}, \frac{\vec{b}}{2} \right\rangle$$

단, 패키지에 `[innerproductbracket]` 옵션을 주었을 때와 달리 별표 붙은 명령을 지원하지는 않는다.

3.5 수열, 미분, 적분

`\ndv`, `\pdv`

사용법은 다음과 같다.

`\ndv{x}`, `\ndv[2]{x}`, `\ndv{y}{x}`, `\ndv[2]{y}{x}`,
 $\hookrightarrow$ `\ndv{z}{x}{y}`
`\pdv{x}`, `\pdv[2]{x}`, `\pdv{y}{x}`, `\pdv[2]{y}{x}`,
 $\hookrightarrow$ `\pdv{z}{x}{y}`

$$\frac{d}{dx}, \frac{d^2}{dx^2}, \frac{dy}{dx}, \frac{d^2y}{dx^2}, \frac{d^2z}{dxdy}$$

$$\frac{\partial}{\partial x}, \frac{\partial^2}{\partial x^2}, \frac{\partial y}{\partial x}, \frac{\partial^2 y}{\partial x^2}, \frac{\partial^2 z}{\partial x \partial y}$$

`\pdv`는 physics 패키지의 것과 사용법이 동일하다.

`\dint`

인라인 수식에서도 큰 적분 기호를 사용하게 해주는 명령이다. 반면 `\int` 명령은 변경 없이 그대로 두었다. 이 명령은 `jpnedumathsymbols`에서 제공하는 것이나 이를 수정하여 `\displaystyle`이 이후의 부호에 영향을 주지 않도록 하였다.

`\dint_{a}^b f(x) dx + \int_a^b f(x) dx`

$$\int_a^b f(x) dx + \int_a^b f(x) dx$$

`\dx, \dy, \dz` (J) 다음과 같은 용법에서 dx, dy 앞에 약간의 간격을 주는 명령이 정의되어 있다.

`\f(x)dx+\f(x)\dx, \dr\ds\dt\du\dx\dy\dz\dtheta,`
 $\rightarrow$ `\dint\dtheta=\theta+\const`

$$f(x)dx + f(x) dx, dr ds dt du dx dy dz d\theta, \int d\theta = \theta + \text{const.}$$

`\const` (J) 적분상수를 나타내는 `\const`는 재정의할 수 있다.

`\renewcommand\const{\ensuremath{C}}`
`\dint\dtheta=\theta+\const`

$$\int d\theta = \theta + C$$

`\defint` (J) 정적분 표현에서 자주 나오는 큰 대괄호. 인자는 3개를 취하는데, 순서가 #1이 아랫 끝이고 #2가 윗끝인 데 주의한다.

`\defint{0}{1}{x}`
 $+\defint{0}{1}{\left(\frac{x}{2}\right)^{\frac{1}{2}}}$

$$\left[x\right]_0^1 + \left[\left(\frac{x}{2}\right)^{\frac{1}{2}}\right]_0^1$$

`\neconcave, \neconvex` (J) $\mathcal{MS}$ -symbols 가운데 있는 `\nearrow`, `\searrow` 등과 비슷하게 다음 명령들이 정의되어 있다. 이 명령은 math mode 내부 또는 외부에서 쓸 수 있다.

`\neconcave \seconcave \seconvex \neconvex`



`\joinedumathsymbols`에서는 별표를 붙여서 약간의 수직 간격을 줄 수 있게 하고 있지만 사실상 불필요한 기능이다.

이 부호를 쓰기 위해서 주로 `mnsymbols`에 의존해 온 경우가 있을 것이다. 그러나 $\text{\LaTeX}$ 에서 아직은 이 심볼 패키지를 안전하게 사용하기가 어렵다. 특히 `amssymb`와 함께 쓸 수 없기 때문에 주의를 요한다. 적어도 함수 증감을 표시하기 위한 `curved arrow` 몇 개를 마련해두는 이유는 이 때문이다.

`signchart` (J) 함수의 증감표를 작성하려면 다음과 같이 한다. `signchart` 환경의 내부에는 math mode가 되지만 이 환경 자체는 수학 모드 밖에서 불러야 한다. (마치 `align` 환경과 비슷하다.)

`\begin{signchart}{5} x&1&\cdots&2&\cdots&3\\\hline`
 $\rightarrow$ `f(x)&0&\neconcave&1&\neconvex&2 \end{signchart}`

x	1	$\cdots$	2	$\cdots$	3
$f(x)$	0	$\curvearrowright$	1	$\curvearrowright$	2

`\seq, \sequence (J)` 수열의 일반항을 표시할 때 쓴다.

`$\$ \backslash seq \{ a_n \} \$$`

$\{a_n\}$

`\dsum` 합기호. `\sum`을 쓰는 경우에는 인라인 수식과 디스플레이 수식에서 다음과 같은 모양이다.

`$\$ \backslash sum_{i=1}^n a_i \$ \quad \$ \backslash displaystyle \backslash sum_{i=1}^n a_i \$$`

$\sum_{i=1}^n a_i \quad \sum_{i=1}^n a_i$

`\dsum`을 쓰면 다음처럼 나타난다. 텍스트스타일에서는 ‘시그마’의 크기가 조금 줄지만 `\limits` 형식의 상하 첨자로 식자한다.

textstyle: `$\$ \backslash dsum \$ \quad \$ \backslash dsum_{i=1}^n a_i \$ \backslash$`
displaystyle: `$\$ \backslash displaystyle \backslash dsum \$ \quad$`
 $\hookrightarrow$ `$\$ \backslash displaystyle \backslash dsum_{i=1}^n a_i \$$`

textstyle: $\sum \sum_{i=1}^n a_i$
displaystyle: $\sum \sum_{i=1}^n a_i$

`\dprod` 마찬가지로 방식으로 곱기호도 정의해두었다. `\prod`는 다음과 같은 모양이다.

`$\$ \backslash prod_{i=1}^n a_i \$ \quad \$ \backslash displaystyle \backslash prod_{i=1}^n a_i \$$`

$\prod_{i=1}^n a_i \quad \prod_{i=1}^n a_i$

`\dprod`를 쓰면 다음처럼 나타난다. 텍스트스타일에서는 ‘파이’의 크기가 조금 줄지만 `\limits` 형식의 상하 첨자로 식자한다.

textstyle: `$\$ \backslash dprod \$ \quad \$ \backslash dprod_{i=1}^n a_i \$ \backslash$`
displaystyle: `$\$ \backslash displaystyle \backslash dprod \$ \quad$`
 $\hookrightarrow$ `$\$ \backslash displaystyle \backslash dprod_{i=1}^n a_i \$$`

textstyle: $\prod \prod_{i=1}^n a_i$
displaystyle: $\prod \prod_{i=1}^n a_i$

3.6 합동, 닮음, 평행

`\equiv` 합동을 나타내는 표준 $\text{\LaTeX}$ 부호인 `\cong` $\cong$ 를 쓰지 않고 $\equiv$ 로 이를 대응한다. 이를 위한 별도의 조치를 취하지 않았다.

`\sim` 닮음을 나타내는 $\sim$ (`\sim`) 대신, 일본과 우리나라에서만 사용하는 부호가 있다. 이를 `\similar`로 표현한다.

```
$\triangle\AA\BB\CC \similar \triangle\AA'\BB'\CC'$
```

$$\triangle ABC \sim \triangle A'B'C'$$

이 명령 자체는 `jpedumathsymbols`에서 제공하지만 `koredumath`는 이를 완전히 다시 썼다. $\text{\XeTeX}$ 또는 $\text{\LuaTeX}$ 에서는 Noto Serif KR 폰트로부터 이 부호를 취하게 되어 있는 것이 디폴트이다. 한편 그림으로 이를 식자하게 할 수 있는데 이를 위해서는 `[similarsignfig]` 옵션을 주어야 한다. $\text{\pdfLaTeX}$ 으로는 이 방식으로 사용한다. 한편, Noto Serif KR이 아닌 다른 폰트로부터 이 부호를 취하려면 `[similarsignfont=<font>]` 형식의 옵션을 `koredumath`에 부여하여야 한다.

`\parallel` (J) 평행을 표시하는 기호는 $\text{\LaTeX}$ 표준 명령과 다른 모양이다. 원래의 기호를 찍으려면 `\originalparallel`.

```
$\overline{\AA\BB} \parallel \overline{\CC\DD}$ \\\
```

```
$\overline{\AA\BB} \originalparallel \overline{\CC\DD}$
```

$$\overline{AB} \parallel \overline{CD}$$

$$\overline{AB} \parallel\!\!\!\parallel \overline{CD}$$

“평행하지 않다”는 부호인 `\notparallel` $\nparallel$ 도 정의되어 있다.

3.7 등호 부등호

`\neq` 2005년 이전 중·고등학교 수학교과에서 ‘같지 않다’는 부호의 사선을 우하향하게 그렸다(≠). 부등호도 같았다. 현재는 국제 표준에 맞게 우상향으로 긋는다(≠). 특이하게 위에 소개한 $\nparallel$ 의 경우에만 사선이 우하향으로 그어져 있기 때문에 착각의 여지가 있다.

`\geq`, `\geqq` 크거나 같다, 작거나 같다는 의미하는 부호를 일본에서는 (여전히) $\geq$, $\leq$ 로 표현한다.
`\leq`, `\leqq` 우리나라에서도 2015년 이전에는 이 부호를 사용했다. 2015년 이후 $\geq$, $\leq$ 를 쓴다.

3.8 간격, 좌표, 괄호

`\comma` (J) 나열에서 쉼표 다음에 약간의 간격을 추가한다.

```
$a_1\comma a_2\comma a_3\comma \ldots a_n$
```

$$a_1, a_2, a_3, \dots a_n$$

`\pair`, `\triplet`, `\quadruplet` (J) 평면좌표, 공간좌표 등을 표현한다. 인자로 주어지는 항목들을 쉼표로 배열할 때의 간격 문제와 괄호 크기 문제를 조절해준다.

```
\pair{1}{\frac{1}{2}} \triplet{1}{\frac{1}{2}}{3}
↪ \quadruplet{1}{\frac{1}{2}}{3}{4}$
```

$$\left(1, \frac{1}{2}\right) \left(1, \frac{1}{2}, 3\right) \left(1, \frac{1}{2}, 3, 4\right)$$

별표를 붙이면 괄호 크기를 조절해주지 않는다. 한편 이 명령들은 좌표뿐 아니고 일반적으로 둘 이상의 항목의 나열에 괄호가 요구되는 상황에서도 쓸 수 있다.

\eparen 하나의 항목만이 있을 경우에 괄호 크기를 조절해준다.

```
\eparen{2}, \eparen{\frac{1}{2}}$
```

$$(2), \left(\frac{1}{2}\right)$$

\braket 어떤 내용을 양편의 괄호로 묶으려 할 때 쓰는 명령이다. 첫 인자는 1, 2, 3 중의 하나이고 1은 괄호(parenthesis), 2는 중괄호, 3은 대괄호이다.

```
\braket{1}{\frac{2}{3}},
↪ \braket{3}{a+\braket{2}{b+\frac{c+d}{e}}}^{\frac{1}{2}}$
```

$$\left(\frac{2}{3}\right), \left[a + \left\{b + \frac{c+d}{e}\right\}\right]^{\frac{1}{2}}$$

\intvl 열린 구간 또는 닫힌 구간을 나타낼 때 쓴다.

```
\intvl[{-1}{2}), \intvl[{a}{b}], \intvl[{-1}{0}),
↪ \intvl[{0}{\frac{1}{2}}]$
```

$$[-1, 2), [a, b], (-1, 0), \left(0, \frac{1}{2}\right]$$

\abs 절댓값. \frac이 쓰일 때에 좌우 \vert의 길이를 조절해준다.

```
\[\abs{\frac{1}{2}}\quad\abs*{\frac{1}{2}}\]
```

$$\left|\frac{1}{2}\right| \quad \left|\frac{1}{2}\right|$$

3.9 집합과 논리

\emptyset $\LaTeX$ 의 표준 \emptyset ($\emptyset$) 대신 $\varnothing$ 을 \emptyset 명령으로 식자되게 했다.

```
\emptyset$, \origemptyset$
```

$$\varnothing, \emptyset$$

\intersection, \union ($\cap, \cup$) \intersection은 \cap과 같고 \union은 \cup과 같다.

```
$A\intersection B\quad A\union B$
```

$$A \cap B \quad A \cup B$$

\set (J) 조건제시법으로 집합을 표현할 때 쓴다. 패키지 옵션으로 [setcolon]을 주어 로드하면 |가 :으로 나타난다.

```
\set{x}{x\geqq\frac{1}{2}}$
```

$$\left\{ x \mid x \geq \frac{1}{2} \right\}$$

\setbrk 앞서 보인 \braket 명령을 이용하여 원소나열법으로 집합을 표현하면 다음과 같다.

```
\setbrk{1\comma 2\comma 3\comma \ldots}$
```

$$\{1, 2, 3, \dots\}$$

\lto, \lfrom, \iff (J) 보기를 보면 간단히 알 수 있다.

```
$A\lto B\qqquad C\lfrom D\qqquad E\iff G$
```

$$A \implies B \quad C \impliedby D \quad E \iff G$$

3.10 cases

ecases, simul koredumath에서 조금 수정하였다. ecases의 디폴트는 align*이다.

```
\begin{ecases}{f(x)}
x&\qqquad\condition{$x\geqq0$}\\
-x&\qqquad\condition{$x<0$}
\end{ecases}
```

$$f(x) = \begin{cases} x & (x \geq 0) \\ -x & (x < 0) \end{cases}$$

더 다양한 사용법은 jpnedumathsymbols 패키지 문서를 볼 것. simul 환경은 연립 방정식에 적용한다. (gather*가 디폴트)

```
\begin{simul}
2x+2y=0\\
x-y=0
\end{simul}
```

$$\begin{cases} 2x + 2y = 0 \\ x - y = 0 \end{cases}$$

이 두 환경은 amsmath의 align*처럼 text mode에서 입력하여야 한다.

3.11 행렬

\matI 항등행렬을 식자한다. 기본은 2×2 이고 옵션을 부여할 수 있다.

```
$\matI{p}$, $\matI{B}[3]$
```

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{Bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{Bmatrix}$$

첫 번째 인자는 예컨대 `pmatrix`에 대해서 `p`, `bmatrix`에 대해서 `b`와 같은 식으로 부여한다. 두 번째 옵션 인자는 숫자이고 예컨대 `n`이면 $n \times n$ 항등행렬을 식자한다.

`\thrthrmatrix` 3×3 정사각행렬을 첨자를 붙여 식자한다.

`$\thrthrmatrix[p]{a}$`

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

옵션 인자는 예컨대 `pmatrix`에 대해서 `p`, `bmatrix`에 대해서 `b`와 같은 식이고, 필수 인자는 첨자가 붙을 문자를 지정하는 것이다.

3.12 그밖의 기호

`\degree (J)` 각도의 ‘도’

`$30\degree$`

30°

`\arc (J)` 호.

`$\arc{\AA\BB}$, $\arc{\AA\BB\CC}$`

$\widehat{AB}, \widehat{ABC}$

`\combination, etc. (J)` 순열(`\permutation`), 조합(`\combination`), 중복순열(`\repeatedpermutation`), 중복조합(`\homogeneous`)

`$\permutation{n}{r}+ \combination{n}{r}+`

$\hookrightarrow \repeatedpermutation{n}{r}+ \repeatedcombination{n}{r}$$

$\hookrightarrow \$\homogeneous{n}{r}$$

${}_nP_r + {}_nC_r + {}_n\Pi_r + {}_nH_r \quad {}_nH_r$

`\Re, \Im (J)` 별도의 캘리그래피 폰트를 쓰지 않는 부호.

`$\Re z+\Im z+\originalRe z+\originalIm z$`

$\operatorname{Re} z + \operatorname{Im} z + \Re z + \Im z$

`\conj, \cml (J)` `\conj`는 `\conjugate`와 같다. `\cml`은 `\complement`와 같다.

`$\conjugate{a}+\conjugate{b}=\conj{a}+\conj{b}$\\`

`$\complement{a}+\complement{b} = \cml{a}+\cml{b}$`

$$\overline{a} + \overline{b} = \overline{a} + \overline{b}$$

$$\overline{a} + \overline{b} = \overline{a} + \overline{b}$$

두 명령은 문자 위에 놓이는 bar의 높이와 길이가 조금 다르다.

`\ccirc` 이 명령은 Kriss의 `intothemath` 패키지에서 가져온 것이다.

$$f \circ g \quad f \circ g$$

`\inverse()`, `\invrs` 역함수 또는 역행렬을 식자하려 할 때.

$$f^{-1}, A^{-1}$$

$$f^{-1}, A^{-1}$$

`\expectedvalue()` 기댓값.

$$\mathbb{E}(P)$$

만약 `[mathbb]`가 주어져 있지 않다면 `\mathbf`로 식자된다.

4 실험적 기능

이 절에서는 실험적(experimental) 기능에 대하여 기술한다. 이 기능은 그 동작의 안정성이 충분히 확인되지 않은 것으로서, 사용자 자신의 책임 하에 시험해볼 수 있다.

4.1 첨자의 위치

`\stdsupersub`, `\varsupersub` 이것은 `jpnedumathsymbols`에 있는 것이 아니다. 수학 교재를 작성하다 보면 $\text{\LaTeX}$ 의 기본 첨자 위치가 익숙하지 않을 때가 있다고 한다. 이를 위하여 `\varsupersub`과 `\stdsupersub` 선언을 마련했다.

$$\text{\varsupersubstep}[3]$$

$$\text{\stdsupersub} a^n \text{\comma} a_x^y \text{\quad} \text{\varsupersub} a^n \text{\comma} a_x^y$$

$$a^n, a_x^y \quad a^n, a_x^y$$

`\varsupersubstep` `\varsupersubstep` 명령으로 올리거나 내리는 길이를 조절할 수 있는데

$$\text{\varsupersubstep}[\langle n \rangle]$$

의 $\langle n \rangle$ 에 해당하는 값을 0에서 4까지 줄 수 있다. 숫자가 커질수록 올리거나 내리는 길이가 길어진다. 기본값은 2이다.

이 기능이 “실험적”인 이유는 폰트에 따라 길잇값이 미묘하게 변할 수 있어서다. 만약 이 기능을 활성화하고자 한다면 `\varsupersubstep`을 한 번 선언하여야 한다. 일반적으로 첨자 위치를 변경하는 것은 권장하지 않는다.

	<code>\stdsupersub</code>	<code>\varsupersub</code>
<code>(\varsupersubstep)</code>	a^2, b^2, c^2	a^2, b^2, c^2
	a_2, b_2, c_2	a_2, b_2, c_2
<code>\varsupersubstep[1]</code>	a^2, b^2, c^2	a^2, b^2, c^2
	a_2, b_2, c_2	a_2, b_2, c_2
<code>\varsupersubstep[2]</code>	a^2, b^2, c^2	a^2, b^2, c^2
	a_2, b_2, c_2	a_2, b_2, c_2
<code>\varsupersubstep[3]</code>	a^2, b^2, c^2	a^2, b^2, c^2
	a_2, b_2, c_2	a_2, b_2, c_2
<code>\varsupersubstep[4]</code>	a^2, b^2, c^2	a^2, b^2, c^2
	a_2, b_2, c_2	a_2, b_2, c_2

4.2 equation의 번호

이것은 **KTUG** **게시판의 글**에 ‘남박사’께서 붙인 답글로 요청하신 것이다. 원하는 결과를 예문 1에 나타내었다.

$$a_n = \sum_{k=1}^n a_k - \sum_{k=1}^{n-1} a_k \quad \dots\dots\dots \textcircled{1}$$

$$= nS_n - (n-1)S_{n-1} (n \geq 2)$$

위에서 $a_1 = S_1$, $a_2 = 2S_2 - S_1$, $a_3 = 3S_3 - 2S_2$ 를 $a_1 + a_3 \geq 2a_2$ 에 대입하여 정리하면,

$$S_1 + S_3 \geq 2S_2$$

따라서 $n = 2$ 일 때 성립한다.

$$S_{k-1} + S_{k+1} \geq S_k \quad \dots\dots\dots \textcircled{2}$$

이 성립할때 2이상의 자연수 k 에 대하여 $\textcircled{2}$ 가 성립함을 가정하면,

$$S_k + S_{k+2} \geq S_{k+1} \quad \dots\dots\dots \textcircled{3}$$

$\textcircled{3}$ 도 성립하므로, 수학적 귀납법에 의하여 모든 자연수 k 에 대하여 성립한다.

《예문》 1

원숫자 수식 번호

수식 번호를 원문자로 한다면 다음과 같이 될 것이다. `[eqnformat=circ]` 옵션을 부여하는 것이다.

```
\usepackage[
  eqnformat=circ
]{koredumath}
```

```
\begin{equation}\label{eq:1}
S_{k-1} + S_{k+1} \geq S_k \ \keepndots
\end{equation}
이 성립할때 $2$이상의 자연수 $k$에 대하여 \eqref{eq:1}\이 성립함을 가정하면,
\begin{equation}\label{eq:2}
S_k + S_{k+2} \geq S_{k+1} \ \keepndots
\end{equation}
\eqref{eq:2}도 성립하므로, 수학적 귀납법에 의하여 모든 자연수 $k$에 대하여 성립한다.
```

$$S_{k-1} + S_{k+1} \geq S_k \quad \textcircled{1}$$

이 성립할때 2 이상의 자연수 k 에 대하여 ①이 성립함을 가정하면,

$$S_k + S_{k+2} \geq S_{k+1} \quad \textcircled{2}$$

②도 성립하므로, 수학적 귀납법에 의하여 모든 자연수 k 에 대하여 성립한다.

원숫자의 모양은 `\koredu_eqnmodifier`라는 매크로가 담당한다. 기본값은 KTUG 사설저장소를 통하여 설치할 수 있는 `kocircnum` 패키지의 `\tikzcircnum` 매크로를 이용하는 것이다. pdf $\LaTeX$ 에서도 무사히 컴파일되게 하기 위한 조치였다. 원숫자 수식 번호를 쓸 때에 주의하여야 할 점은 수식 번호가 장 번호나 절 번호와 연결되어 있으면 좋지 않다는 것이다. 그래서

```
\counterwithin{equation}{chapter}
```

와 같은 명령을 주면 곤란하다.

`\reseteqn` 이 명령은 `equation`의 카운터를 0으로 되돌린다. ‘eqn’이란 ‘equation’의 줄인 말이 아니고 ‘equation number’를 줄인 말이다.

점선이 붙는 수식 번호

수식의 끝에서 수식 번호까지 점선으로 채우는 관행이 있다고 한다. 조판상 그다지 좋아 보이지 않는 것이기는 하지만 교육 현장에서는 많이 쓴다고 하니 어쨌든 구 현해보기로 하자. 현재 버전에서는 오직 `equation`에 대해서만 동작하게 되어 있고 `align`과 같은 여러 줄 수식에서는 적용되지 않는다.

`\keepndots` 점선이 붙는 수식의 마지막 위치에 `\keepndots`라는 매크로를 반드시 지시하여야 한다. 이 매크로가 없으면 점선이 나타나지 않는다. 이 매크로는 `[eqnformat=dots]`가 주어지지 않았을 때는 아무 역할도 하지 않는다.

`[eqnformat=dots]` 옵션을 주면 다음과 같이 된다. 반드시 두 번 이상 컴파일해야 원하는 결과를 얻을 수 있다.

```
\usepackage[eqnformat=dots]{koredumath}

\begin{equation}\label{eq:1}
S_{k-1} + S_{k+1} \geq S_k \ \keepndots
\end{equation}
이 성립할때 $2$이상의 자연수 $k$에 대하여 \eqref{eq:1}\이 성립함을 가정하면,
\begin{equation}\label{eq:2}
S_k + S_{k+2} \geq S_{k+1} \ \keepndots
\end{equation}
```


$$S_{k-1} + S_{k+1} \geq S_k \quad \cdots \cdots \cdots \textcircled{1}$$

이 성립할때 2 이상의 자연수 k 에 대하여 $\textcircled{1}$ 이 성립함을 가정하면,

$$S_k + S_{k+2} \geq S_{k+1} \quad \cdots \cdots \cdots \textcircled{2}$$

$\textcircled{2}$ 도 성립하므로, 수학적 귀납법에 의하여 모든 자연수 k 에 대하여 성립한다.

[eqnformat=none]을 부여하면 이 모든 설정을 무시하게 되므로 L^AT_EX 기본과 동일하게 될 것이다. 이 패키지의 [eqnformat] 관련 기능은 실험적인 것이므로 none을 기본값으로 하고 있다.

```
\usepackage[eqnformat=none]{koredumath}

\begin{equation}\label{eq:1}
S_{k-1} + S_{k+1} \geq S_k \quad \keeqndots
\end{equation}
이 성립할때 $2$이상의 자연수 $k$에 대하여 \eqref{eq:1}\이 성립함을 가정하면,
\begin{equation}\label{eq:2}
S_k + S_{k+2} \geq S_{k+1} \quad \keeqndots
\end{equation}
\eqref{eq:2}도 성립하므로, 수학적 귀납법에 의하여 모든 자연수 $k$에 대하여 성립한다.
```

$$S_{k-1} + S_{k+1} \geq S_k \quad (1)$$

이 성립할때 2 이상의 자연수 k 에 대하여 (1)이 성립함을 가정하면,

$$S_k + S_{k+2} \geq S_{k+1} \quad (2)$$

(2)도 성립하므로, 수학적 귀납법에 의하여 모든 자연수 k 에 대하여 성립한다.

5 마치는 말

이 패키지는 실제 대학교재의 조판을 위해 실용적 목적으로 작성하였고 **KTUG 게시판:260097**로 선을 보였으며 2023년 마지막 **공주대학교 문서작성워크숍**을 통하여 발표하였다. 이 날의 발표자료는 이 매뉴얼에 담고 있지 않은 동기나 배경을 좀더 상세하게 설명하고 있다.

패키지 사용에 있어 문제를 발견하거나 개선에 대한 제안을 환영한다. KTUG 게시판을 통하여 대화하고 문제를 해결해나가고자 하므로 게시판에 글을 올려주시기를 바란다.

이 패키지가 구현하는 수식이 바람직한 것은 아니다. 교육용 수학식이 더 보편적이고 받아들일 만한 기준을 갖기를 원하지만 현장에서 실무에 임하는 편집자와 조판자에게 당장 필요한 것을 제공한다는 점에서 무의미한 작업은 아니었을 것이다.

謝辭

공주대학교 문서작성워크숍 2023의 발표를 통하여 이 패키지를 전면적으로 검토하고 개선할 수 있는 기회가 되었다. 워크숍을 주최하고 진행하신 공주대학교 경제학과와

주관기관인 한국텍학회 및 한글텍사용자그룹에 감사한다. 오랜 역사를 자랑하는 공주대학교 문서작성워크숍이 이로써 마지막이 된다는 데 아쉬운 마음을 표현해둔다. KTUG 게시판에 공개하였을 때 관심을 보여주고 제안과 기여를 해주신 분들께도 감사한다.

찾아보기

【 A 】

<code>\AA</code>	1
<code>\abs</code>	7
<code>align</code>	4
<code>align*</code>	8
<code>amsmath</code>	2, 8
<code>amssymb</code>	1, 4
<code>\arc</code>	9

【 B 】

<code>\BB</code>	1
<code>\braket</code>	7

【 C 】

<code>\C</code>	1
<code>\cap</code>	7
<code>\ccirc</code>	10
<code>\cml</code>	9
<code>\combination</code>	9
<code>\comma</code>	6
<code>\complement</code>	9
<code>\cong</code>	6
<code>\conj</code>	9
<code>\conjugate</code>	9
<code>\const</code>	4
<code>\cup</code>	7
<code>\cvec</code>	2
<code>\cvec*</code>	2
<code>[cvecbracket]</code>	3

【 D 】

<code>\defint</code>	4
<code>\degree</code>	9
<code>\dint</code>	3
<code>\displaystyle</code>	3
<code>\dprod</code>	5
<code>\dsum</code>	5
<code>\dx</code>	4
<code>\dy</code>	4
<code>\dz</code>	4

【 E 】

<code>ecases</code>	8
<code>\emptyset</code>	7
<code>\eparen</code>	7
<code>[eqnformat]</code>	1, 14
<code>[eqnformat=circ]</code>	11
<code>[eqnformat=circdots]</code>	13
<code>[eqnformat=dots]</code>	12
<code>[eqnformat=dotscirc]</code>	13
<code>[eqnformat=none]</code>	14
<code>\equiv</code>	6
<code>\expectedvalue</code>	10

【 F 】

<code>[fleqn]</code>	13
<code>\frac</code>	2, 7

【 G 】

<code>gather*</code>	8
<code>\geq</code>	6
<code>\geqq</code>	6

【 H 】

<code>\homogeneous</code>	9
---------------------------------	---

【 I 】

<code>\iff</code>	8
<code>\Im</code>	9
<code>\innerproduct</code>	3
<code>[innerproductbracket]</code>	3
<code>\inp</code>	3
<code>\inpbracket</code>	3
<code>\int</code>	3
<code>\intersection</code>	7
<code>intothemath</code>	10
<code>\intvl</code>	7
<code>\inverse</code>	10
<code>\invrs</code>	10

【 J 】

<code>jpnedumathsymbols</code>	1-4, 6, 8
<code>\jpnsqrt</code>	2

