

제 3 장

수식 조판하기

이제 준비가 됐다! 이 장에서 우리는 \TeX 의 주된 강점인 수식 조판을 공략할 것이다. 하지만, 이 장은 그저 수박 겉핥기라는 점을 일러 둔다. 여기서 설명하는 것만으로도 많은 이들에게 충분하겠지만, 원하는 수식 조판의 답을 여기서 찾지 못했다고 좌절하진 말기 바란다. \LaTeX ¹에 해답이 있을 확률이 높다.

제 1 절 개괄

\LaTeX 은 수학적 수식 조판을 위한 특별한 모드를 갖고 있다. 수학적 수식은 단락 안에서 행중(`inline`) 수식으로 조판될 수도 있고, 별도의 단락으로 조판될 수도 있다. 단락 안의 행중 수식 텍스트는 `\(`와 `\)`, `$`와 `$`, 또는 `\begin{math}`와 `\end{math}` 사이에 들어간다.

Add a squared and b squared to get c squared. Or, using a more mathematical approach: $c^2 = a^2 + b^2$

Add a squared and b squared to get c squared. Or, using a more mathematical approach: $c^2 = a^2 + b^2$

\TeX is pronounced as $\tau\epsilon\chi$.
100 m³ of water
This comes from my \heartsuit

\TeX is pronounced as $\tau\epsilon\chi$.
100 m³ of water
This comes from my \heartsuit

별도 단락으로 큰 수학 기호를 사용하는 방정식이나 공식을 식자하려면 실제로 단락을 나누는 것보다 수식 보여주기(`display`) 하는 것이 바람직하다.

¹미국수학회는 \LaTeX 의 강력한 확장판을 만들어 냈다. 이 장에서 많은 예제들은 이 확장판을 쓴다. 최근에 나온 모든 \TeX 배포판들은 이를 제공한다. 만약 빠져 있다면 `macros/latex/required/amslatex`에 가 보자.

이렇게 하기 위해서, 이들을 `\[`와 `\]`안에 넣거나, `\begin{displaymath}`와 `\end{displaymath}` 사이에 넣는다.

```
Add $a$ squared and $b$ squared
to get $c$ squared. Or, using
a more mathematical approach:
\begin{displaymath}
c^2=a^2+b^2
\end{displaymath}
or you can type less with:
\[a+b=c\]
```

Add a squared and b squared to get c squared. Or, using a more mathematical approach:

$$c^2 = a^2 + b^2$$

or you can type less with:

$$a + b = c$$

수식 번호를 L^AT_EX이 매겨주기를 원한다면, `equation` 환경을 쓸 수 있다. 그러면 수식 번호에 `\label`을 붙이고 `\ref` 또는 `amsmath` 패키지에 있는 `\eqref` 명령을 써서 본문 안 다른 곳에서 이 레이블을 참조할 수 있다.

```
\begin{equation} \label{eq:eps}
\epsilon > 0
\end{equation}
From (\ref{eq:eps}), we gather
\ldots{}From \eqref{eq:eps} we
do the same.
```

$$\epsilon > 0 \quad (3.1)$$

From (3.1), we gather ... From (3.1) we do the same.

행중(inline) 수식과 수식 보여주기(display) 수식 사이에는 조판 스타일에 차이가 있다는 점을 유의하자.

```
 $\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

수학 모드와 텍스트 모드 사이에는 차이점이 있다. 예를 들어서, 수학 모드 안에서는,

1. 빈 칸과 줄바꿈은 대부분 어떤 의미도 갖지 않는다. 수식의 빈 칸은 수학적 표현으로부터 논리적으로 나오거나 `\,`, `\quad` 또는 `\qquad`와 같은 특별한 명령으로 지정해야 한다.
2. 빈 줄은 허용되지 않는다. 공식마다 단 한 문단 썬이다.

3. 각 글자는 변수명으로 간주되며 변수로 조판될 것이다. 보통 텍스트(정상적인 곧게 선 글꼴과 일반적인 간격)를 공식 안에서 조판하고자 한다면 텍스트를 `\textrm{...}` 명령 안에 넣어야 한다 (49페이지의 7도 참조할 것).

```
\begin{equation}
\forall x \in \mathbf{R}:
\quad x^2 \geq 0
\end{equation}
```

$$\forall x \in \mathbf{R}: \quad x^2 \geq 0 \quad (3.2)$$

```
\begin{equation}
x^2 \geq 0 \quad \text{for all } x \in \mathbf{R}
\end{equation}
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbf{R} \quad (3.3)$$

수학자들은 기호문자 선택에 무척 까다로울 수 있다. 위의 경우 실수 집합을 나타내는 데 굵은 글씨 **R**이 아니라 ‘블랙보드 볼드’(blackboard bold)를 쓰는 것이 관행인데 이것은 `amsmath` 또는 `amssymb` 패키지에 있는 `\mathbb{R}` 명령을 써서 얻을 수 있다.

그러므로 예제 마지막 줄은 다음과 같이 한다.

```
\begin{displaymath}
x^2 \geq 0 \quad \text{for all } x \in \mathbb{R}
\end{displaymath}
```

$$x^2 \geq 0 \quad \text{for all } x \in \mathbb{R}$$

제 2 절 수학 모드에서 묶기

대부분 수학 모드 명령들은 바로 다음 글자에 대해서만 유효하기 때문에, 만약 명령이 여러 글자에 영향을 미치기를 원한다면, 이들을 중괄호를 써서 묶어 주어야 한다: `{...}`.

```
\begin{equation}
a^{x+y} \neq a^{x+y}
\end{equation}
```

$$a^x + y \neq a^{x+y} \quad (3.4)$$

제 3 절 수식 구성하기

이 절에서는 수식 조판에 쓰이는 가장 중요한 명령들을 설명한다. 수학 기호 조판을 위한 명령의 자세한 목록을 보려면 53 페이지에 있는 제 10 절을 살펴 보라.

그리스 소문자는 `\alpha`, `\beta`, `\gamma`, ...와 같이 입력하고 대문자들은 `\Gamma`, `\Delta`, ...와 같이 입력한다.²

`\lambda, \xi, \pi, \mu, \Phi, \Omega`

$\lambda, \xi, \pi, \mu, \Phi, \Omega$

윗첨자와 아랫첨자는 `^` 그리고 `_` 글자를 써서 지정할 수 있다.

`a_{1} \quad x^{2} \quad`
`$e^{-\alpha t}$ \quad`
`a^{3}_{ij} \quad`
`$e^{x^2} \neq e^{x^2}$`

$a_1 \quad x^2 \quad e^{-\alpha t} \quad a_{ij}^3$
 $e^{x^2} \neq e^{x^2}$

제곱근은 `\sqrt`로 만든다. n차 제곱근은 `\sqrt[n]`로 만들어진단다. 제곱근 기호의 크기는 L^AT_EX가 자동으로 판단한다. 근호만이 필요할 경우라면, `\surd`를 쓰도록 하자.

`\sqrt{x} \quad`
`$\sqrt{x^2+\sqrt{y}}$ \quad`
`\quad $\sqrt[3]{2}$ \quad`
`$\surd[x^2 + y^2]$`

$\sqrt{x} \quad \sqrt{x^2 + \sqrt{y}} \quad \sqrt[3]{2}$
 $\sqrt{x^2 + y^2}$

`\overline`과 `\underline` 명령은 수식의 바로 위나 아래에 수평선을 긋는다.

`$\overline{m+n}$`

$\overline{m+n}$

`\overbrace`와 `\underbrace`는 수식의 위나 아래에 긴 수평 중괄호를 만든다.

`$\underbrace{a+b+\cdots+z}_{26}$`

$\underbrace{a+b+\cdots+z}_{26}$

작은 화살표나 물결표 기호와 같은 수학적 액센트를 변수에 더하기 위해 서, 53 페이지에 있는 표 3.1에 있는 명령을 쓸 수 있다. `\widehat` 그리고 `\widetilde` 명령으로 여러 글자에 걸쳐서 넓은 모자 기호나 물결표를 만들 수 있다. `prime` 명령은 ' 기호를 만든다.

`\begin{displaymath}`
`y=x^2 \quad y'=2x \quad y''=2`
`\end{displaymath}`

$y = x^2 \quad y' = 2x \quad y'' = 2$

²L^AT_EX 2_ε에는 대문자 알파가 정의되어 있지 않은데, 보통 모양 로마자 A와 똑같이 보이기 때문이다. 새로운 수학 기호화가 이루어지고 나면, 상황은 바뀔 것이다.

벡터는 종종 변수의 위에 작은 화살표 부호를 더하여 나타낸다. 이는 `\vec` 명령으로도 할 수 있다. 두 명령 `\overrightarrow`와 `\overleftarrow`는 A 에서 B 로 가는 벡터를 표시할 때 쓸모가 있다.

```
\begin{displaymath}
\vec a\quad\overrightarrow{AB}
\end{displaymath}
```

$$\vec{a} \quad \overrightarrow{AB}$$

보통은 곱셈 연산을 가리키기 위한 도트 기호를 명시적으로 조판하지 않는다. 하지만 때때로 공식을 그룹으로 묶어서 독자가 알아보기 쉽게 만들기 위해서 이 기호를 쓰기도 한다. 이런 경우에는 `\cdot`를 써야 한다.

```
\begin{displaymath}
v = {\sigma}_1 \cdot {\sigma}_2 \tau_1 \cdot \tau_2
\end{displaymath}
```

$$v = \sigma_1 \cdot \sigma_2 \tau_1 \cdot \tau_2$$

로그(log)와 같은 함수 이름은 대개 이탤릭체나 변수명과 같은 모양이 아닌 곧게 선 글꼴로 조판되며, 따라서 \LaTeX 은 중요한 대부분의 함수 이름을 조판하기 위하여 다음과 같은 명령들을 제공한다.

```
\arccos \cos \csc \exp \ker \limsup \min
\arcsin \cosh \deg \gcd \lg \ln \Pr
\arctan \cot \det \hom \lim \log \sec
\arg \coth \dim \inf \liminf \max \sin
\sinh \sup \tan \tanh
```

```
\[\lim_{x \rightarrow 0}
\frac{\sin x}{x}=1\]
```

$$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

모듈로 함수(modulo function)를 위해서는 두 가지 명령이 있다. 이항 연산자에 해당하는 `\bmod` 명령과 “ $x \equiv a \pmod{b}$ ”와 같은 표현을 위한 `\pmod`가 그것이다.

```
$a\bmod b$\$
$x\equiv a \pmod{b}$
```

$$a \bmod b$$

$$x \equiv a \pmod{b}$$

분모와 분자를 상하로 배열하는 분수는 `\frac{...}{...}` 명령으로 조판된다. 종종 $1/2$ 과 같이 사선을 그어 표시하는 형식이 더 나올 수도 있는데, 적은 양의 ‘분수류’에 대해서는 사선 형식이 더 나아보이기 때문이다.

```

 $\frac{1}{2}$  hours
\begin{displaymath}
\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}
\end{displaymath}

```

 $1\frac{1}{2} \text{ hours}$

$$\frac{x^2}{k+1} \quad x^{\frac{2}{k+1}} \quad x^{1/2}$$

이항 계수나 그와 비슷한 구조를 조판하기 위해서는, `amsmath` 패키지의 `\binom` 명령을 쓸 수 있다.

```

\begin{displaymath}
\binom{n}{k} \quad \mathrm{C}_n^k
\end{displaymath}

```

$$\binom{n}{k} \quad C_n^k$$

이항 관계식에서는 기호를 위 아래에 배열해야 할 때가 있다. `\stackrel`은 두번째 인자를 정상적인 위치에 두고 첫번째 인자로 주어진 기호를 그 윗쪽에 상첨자 크기로 놓는다.

```

\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}

```

$$\int f_N(x) \stackrel{!}{=} 1$$

적분 기호는 `\int`로, 합 기호는 `\sum`으로, 그리고 곱 기호는 `\prod`로만 들어진다. 위끝 아래끝 극한은 윗첨자와 아랫첨자처럼 `^`과 `_` 기호로 만든다.³

```

\begin{displaymath}
\sum_{i=1}^n \quad \int_0^{\pi/2} \quad \prod_{\epsilon}
\end{displaymath}

```

$$\sum_{i=1}^n \quad \int_0^{\pi/2} \quad \prod_{\epsilon}$$

복잡한 수식에 있어서 지수의 위치 조절을 위해 `amsmath`는 두 가지 추가 도구를 제공한다. `\substack` 명령과 `subarray` 환경이다.

```

\begin{displaymath}
\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) =
\sum_{\substack{i \in I \\ 1 < j < m}} Q(i, j)
\end{displaymath}

```

$$\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) = \sum_{\substack{i \in I \\ 1 < j < m}} Q(i, j)$$

$\mathrm{T}_\mathrm{E}\mathrm{X}$ 은 대괄호와 기타 짝맞춤 문자(예를 들어, $\langle \parallel \updownarrow \rangle$)를 위한 모든 종류의 기호를 제공한다. 둥근 괄호(소괄호)와 각진 괄호(대괄호)는 키보드의 해당

³ $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathrm{L}\mathrm{A}\mathrm{T}_\mathrm{E}\mathrm{X}$ 패키지는 여러 줄에 걸친 윗첨자와 아랫첨자를 쓸 수 있다.

키로부터, 그리고 중괄호는 `\{`로 입력할 수 있으나, 그밖의 모든 짝맞춤 기호들은 특별한 명령으로 만들어진단 (예를 들어, `\updownarrow`). 사용할 수 있는 모든 짝맞춤 기호의 목록은 55 페이지에 있는 표 3.8에서 찾아볼 수 있다.

```
\begin{displaymath}
\{a,b,c\}\neq\{a,b,c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

만약 여는 짝맞춤 기호 앞에 `\left` 명령을 놓았거나 닫는 짝맞춤 기호 뒤에 `\right` 명령을 놓았다면, \TeX 은 짝맞춤 기호의 적당한 크기를 자동으로 정할 것이다. `\left`를 쓸 때마다 그 다음에 `\right`로 닫아줘야 한다는 점, 그리고 둘 모두 같은 줄에서 조판이 될 때에만 크기가 적절하게 정해진다는 점을 주의하자. 만약 오른쪽 짝맞춤 기호 위치에 아무것도 쓰지 않아야 한다면 보이지 않는 ‘`\right.`’를 사용하자.

```
\begin{displaymath}
1 + \left( \frac{1}{1-x^2} \right)^3
\end{displaymath}
```

$$1 + \left(\frac{1}{1-x^2} \right)^3$$

어떤 경우 수학 짝맞춤 기호의 올바른 크기를 수동으로 지정할 필요가 있다. 이럴 때는 `\big`, `\Big`, `\bigg` 그리고 `\Bigg` 명령을 접두사처럼 짝맞춤 기호 명령 앞에 붙임으로써 대부분 지정할 수 있다.⁴

```
$\Big( (x+1)(x-1) \Big)^2$\\
$\big(\Big(\bigg(\Bigg(\quad$\\
$\big)\Big)\bigg)\Bigg)\quad$\\
$\big|\Big|\bigg|\Bigg|$\
```

$$\left((x+1)(x-1) \right)^2 \\ \left(\left(\left(\right) \right) \right) \left| \right| \left| \right| \left| \right| \left| \right|$$

점 세 개를 수식 안에 넣기 위한 여러 가지 명령이 있다. `\ldots`는 점들을 베이스라인에 식자하며 `\cdots`는 중앙에 식자한다. 여기에 더해서, 점 세 개를 세로로 놓는 `\vdots`와 점 세 개를 대각선으로 놓는 `\ddots`가 있다. 제 5 절에서 다른 예제를 찾을 수 있다.

```
\begin{displaymath}
x_1, \ldots, x_n \quad \qquad \qquad \\
x_1 + \cdots + x_n
\end{displaymath}
```

$$x_1, \dots, x_n \qquad x_1 + \cdots + x_n$$

⁴이들 명령은 글꼴 크기 변경 명령이 쓰인 경우나 11pt 또는 12pt 선택 사항들이 지정된 경우에는 원하는 대로 작동하지 않는다. 이런 문제점을 바로 잡기 위해서는 `exscale` 또는 `amsmath` 패키지를 쓰도록 하자.

array 환경은 오른쪽 짝맞춤 문자(\right)로 보이지 않게 하는 “.”를 써서 한 개의 큰 묶음표를 가지는 수식을 조판하는 데에도 쓸 수 있다.

```
\begin{displaymath}
y = \left\{ \begin{array}{l}
a & \text{if } d > c \\
b+x & \text{in the morning} \\
1 & \text{all day long}
\end{array} \right.
\end{displaymath}
```

$$y = \begin{cases} a & \text{if } d > c \\ b+x & \text{in the morning} \\ 1 & \text{all day long} \end{cases}$$

tabular 환경처럼 array 환경 안에서도 줄을 그을 수 있다. 예를 들어서 행렬의 원소를 갈라 놓으려면,

```
\begin{displaymath}
\left( \begin{array}{c|c}
1 & 2 \\ \hline
3 & 4
\end{array} \right)
\end{displaymath}
```

$$\left(\begin{array}{c|c} 1 & 2 \\ \hline 3 & 4 \end{array} \right)$$

여러 줄에 걸친 식이나 수식군을 위해서는 eqnarray 환경을 쓸 수 있으며, equation 대신에 eqnarray*를 쓸 수 있다. eqnarray 안에서 각 줄은 하나의 수식 번호를 갖게 된다. eqnarray*는 아무도 번호를 갖지 않는다.

eqnarray와 eqnarray* 환경은 {rc1} 형식의 세 컬럼 표(tabular)와 같이 동작하며, 가운데 칸은 등호, 부등호, 그밖에 적당한 다른 기호를 두는 데 쓴다. \\ 명령으로 줄을 나눈다.

```
\begin{eqnarray}
f(x) & = & \cos x & \\
f'(x) & = & -\sin x & \\
\int_0^x f(y)dy & & & \\
& = & \sin x & \\
\end{eqnarray}
```

$$\begin{aligned} f(x) &= \cos x & (3.5) \\ f'(x) &= -\sin x & (3.6) \\ \int_0^x f(y)dy &= \sin x & (3.7) \end{aligned}$$

등호 양쪽 편의 간격이 좀 넓다는 것을 알아 두자. 다음 예제에서처럼 \setlength\arraycolsep{2pt} 설정으로 이를 줄일 수 있다.

긴 수식은 자동으로 적절하게 나누어지지 않을 것이다. 저자는 어디서 나눌 것인지 들여쓰기를 얼마로 할지 지정해야 한다. 이를 위해서 다음 두 방법이 가장 널리 쓰인다.

```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x &= & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
&& \frac{x^7}{7!} + \cdots
\end{eqnarray}}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \quad (3.8)$$

```
\begin{eqnarray}
\lefteqn{\cos x = 1} \\
& - \frac{x^2}{2!} + \\
& \frac{x^4}{4!} - \\
& \frac{x^6}{6!} + \cdots
\end{eqnarray}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \quad (3.9)$$

`\nonumber` 명령은 L^AT_EX에게 이 수식에 번호를 붙이지 말라고 알려준다.

이런 방법들로 세로로 똑바르게 정렬되어 배열되는 수식을 얻기 어려울 수 있다. `amsmath` 패키지는 더욱 강력한 여러 가지 대안들을 제공한다.

제 6 절 허깨비들

허깨비는 눈에 보이지 않지만 여전히 많은 사람들의 마음 속에 한 자리를 차지하고 있다. L^AT_EX도 다를 바 없다. 눈에 보이지 않는 공백 처리를 위해 허깨비(`phantom`)를 응용할 수 있다.

`^`와 `_`를 써서 텍스트를 세로로 정렬할 때 L^AT_EX에게 맡겨두면 결과가 반드시 원하지대로 나오지 않는 수가 있다. `\phantom` 명령을 쓰면 최종 출력에서 보이지 않는 글자들만큼 빈 공간을 확보할 수 있다. 다음 예제를 보면 쉽게 이해가 될 것이다.

```
\begin{displaymath}
{}^{12}_{\phantom{1}6}\text{C} \\
\quad \quad \quad \text{versus} \\
{}^{12}_{6}\text{C}
\end{displaymath}
```

$${}^{12}_{6}\text{C} \quad \text{versus} \quad {}^{12}_6\text{C}$$

```
\begin{displaymath}
\Gamma_{ij}^{\phantom{ij}k} \\
\quad \quad \quad \text{versus} \\
\Gamma_{ij}^k
\end{displaymath}
```

$$\Gamma_{ij}^{k} \quad \text{versus} \quad \Gamma_{ij}^k$$

제 7 절 수학 글꼴 크기

수학 모드에서 \TeX 은 글자 크기를 문맥에 따라 스스로 선택한다. 예를 들어, 윗첨자는 더 작은 글꼴로 식자된다. 수식의 한 부분을 곧게 선 글꼴로 하고 싶다면 $\text{\texttt{\textbackslashtextrm}}$ 명령을 쓰면 안되는데, $\text{\texttt{\textbackslashtextrm}}$ 명령은 임시로 텍스트 모드로 빠져 나오기 때문에 글꼴 크기 전환 메커니즘이 작동하지 않기 때문이다. 크기 전환 메커니즘을 계속 유지하려면 대신 $\text{\texttt{\textbackslashmathrm}}$ 를 쓰도록 하자. 하지만 $\text{\texttt{\textbackslashmathrm}}$ 은 짧은 범위에서만 잘 동작할 것이다. 공백들은 여전히 전환되지 않을 것이고 액센트가 붙은 문자들에는 통하지 않을 것이다. ⁶

```
\begin{equation}
2^{\text{\textrm{nd}}} \quad \quad \backslash quad
2^{\mathrm{nd}}
\end{equation}
```

$$2^{\text{nd}} \quad 2^{\text{nd}} \quad (3.10)$$

그래도 \LaTeX 에게 올바른 글꼴 크기를 알려줘야 하는 때가 가끔 있다. 수학 모드에서는 다음 네 가지 명령으로 이를 설정한다.

`\displaystyle (123)`, `\textstyle (123)`, `\scriptstyle (123)` and `\scriptscriptstyle (123)`.

스타일을 바꾸면 상하한값이 표시되는 방식도 달라진다.

$$\mathop{\mathrm{corr}}(X,Y)=\frac{\sum_{i=1}^n(x_i-\overline{x})(y_i-\overline{y})}{\biggl[\sum_{i=1}^n(x_i-\overline{x})^2\sum_{i=1}^n(y_i-\overline{y})^2\biggr]^{1/2}}$$

$$\text{corr}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2}}$$

이것은 `\left[\right]`에 의한 것보다 더 큰 대괄호를 만드는 보기이다.
`\biggl`과 `\biggr` 명령은 각각 왼쪽과 오른쪽 대괄호를 만들어준다.

제 8 절 정리, 법칙, ...

수학 문서를 작성할 때, “보조 정리(Lemma)”, “정의(Definition)”, “공리(Axiom)” 류의 구조를 조판하는 방법을 필요로 할 경우가 있다. L^AT_EX은 이에 해당하는

⁶ $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ (amsmath) 패키지를 쓰면 `\textrm` 명령이 크기 변경과 함께 작동하도록 한다.

명령을 지원한다.

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

name 인자는 “정리”를 식별하기 위한 짧은 키워드다. *text* 인자로는 최종 문서에서 인쇄될 “정리”의 실제 이름을 정의한다.

대괄호 안에 있는 인자들은 선택 사항이다. 둘 다 “정리”에서 번호를 매기는 방법을 지정하는 데에 쓰인다. *counter* 인자는 이전에 정의된 “정리”의 명칭(*name*)으로 지정하도록 하라. 그러면 새로 정의된 정리가 이전 “정리”의 일련 번호를 이어받아 매겨질 것이다. *section* 인자는 “정리” 번호가 어떤 장 절 단위의 번호에 연동되도록 할 것인지 지시하는 것이다. 이것을 *section* 으로 하면 새로운 절이 시작될 때 번호를 새로 시작하면서 정리의 번호를 절 번호. 정리번호 형식으로 식자해준다.

전처리부에서 `\newtheorem` 명령을 선언한 후 문서에서 다음 명령을 쓸 수 있다.

```
\begin{name}[text]
This is my interesting theorem
\end{name}
```

설명은 이 정도면 될 것이다. 다음 예제들을 통하여 남아 있을 의문을 없애고 복잡해보이는 `\newtheorem` 환경의 용법을 명확히 하도록 하자.

```
% definitions for the document
% preamble
\newtheorem{law}{Law}
\newtheorem{jury}[law]{Jury}
%in the document
\begin{law} \label{law:box}
Don't hide in the witness box
\end{law}
\begin{jury}[The Twelve]
It could be you! So beware and
see law~\ref{law:box}\end{jury}
\begin{law}No, No, No\end{law}
```

Law 1 *Don't hide in the witness box*

Jury 2 (The Twelve) *It could be you! So beware and see law 1*

Law 3 *No, No, No*

“Jury” 정리는 “Law” 정리와 같은 카운터를 쓰게 설정되었다. 그러므로 “Jury”의 정리 번호는 이전의 “Law” 정리 번호에 이어진다. 정리 환경의 대괄호 안 옵션 인자는 정리의 제목이나 표제를 특정하는 데 쓴다.

```
\flushleft
\newtheorem{mur}{Murphy}[section]
\begin{mur}
If there are two or more
ways to do something, and
one of those ways can result
in a catastrophe, then
someone will do it.\end{mur}
```

Murphy 8.1 *If there are two or more ways to do something, and one of those ways can result in a catastrophe, then someone will do it.*

“머피” 정리는 현재 절 번호와 연동되어 있다. 이 자리에는 chapter나 sub-section과 같은 장절 단위를 사용할 수 있다.

제 9 절 수학 볼드 기호

L^AT_EX에서 두꺼운 기호를 식자하는 것이 쉽지 않다. 아마도 아마추어 조판가들이 이를 남용하려는 경향이 있어서 일부러 그런 것 같다. 글꼴을 바꾸는 명령인 `\mathbf`로 두툼한 글자를 만들 수 있지만, 수학 기호들이 보통 이탤릭인 데 반해 이렇게 하면 로만(곧게 선) 글꼴이 된다. `\boldmath` 명령이 있지만, 이 명령은 수학 모드 밖에서만 쓸 수 있으므로 텍스트 모드로 빠져나가서 사용해야 한다. 그리고 문장부호에도 함께 적용된다.

```
\begin{displaymath}
\mu, M \quad \mathbf{M} \quad \mu, M
\end{displaymath}
```

$$\mu, M \quad \mathbf{M} \quad \mu, M$$

셈표까지 굵은 기호가 되어 있는데, 이것은 원하는 바가 아닐 수도 있다는 것에 유의하자.

`amsbsy` 패키지(`amsmath` 패키지에 포함되어 있다), 또는 `tools` 패키지 묶음에 들어 있는 `bm` 패키지를 쓰면 `\boldsymbol` 명령으로 훨씬 쉽게 이 문제를 해결하게 해준다.

```
\begin{displaymath}
\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}
\end{displaymath}
```

$$\mu, M \quad \boldsymbol{\mu}, \boldsymbol{M}$$

이밖에 `amsbsy` 패키지의 `\pmb` 명령은 이른바 “가난한 사람을 위한 굵은 수학 기호”(Poorman’s bold)를 이용하여 원하는 대부분의 문자를 볼드로 만들어준다. 이 ‘가짜 볼드’가 마음에 들지 않는다면 볼드 수학 기호를 포함하고 있는 수학 글꼴 묶음을 찾아보는 방법도 있다. 예를 들어 `mathdesign`의 글꼴들 가운데 볼드 수학 기호를 포함하고 있는 것도 있다. 그러나 이 때는 익숙한 CM의 수학 글꼴을 사용할 수 없게 된다. 다음은 ‘가짜 볼드’를 사용한 예이다.

```
\begin{displaymath}
\mu, M \quad\quad
\pmb{\mu}, \pmb{M}
\end{displaymath}
```

$\mu, M \quad\quad \pmb{\mu}, \pmb{M}$
--

제 10 절 수학 기호 일람

다음 표들은 수학 모드에서 사용할 수 있는 기호들을 보여준다.

표 3.12-3.16에 열거된 기호를 사용하려면⁷ `amssymb` 패키지를 문서의 전처리부에 선언하여야 하고 AMS 수학 글꼴이 시스템에 설치되어 있어야 한다. 만약 AMS 패키지와 글꼴이 설치되어 있지 않다면 `macros/latex/amslatex`을 찾아보라. 기호문자에 대한 보다 완전한 목록은 `info/symbols/comprehensive`에서 볼 수 있다.

표 3.1: 수학 모드 악센트

\hat{a}	<code>\hat{a}</code>	\check{a}	<code>\check{a}</code>	\tilde{a}	<code>\tilde{a}</code>	\acute{a}	<code>\acute{a}</code>
\grave{a}	<code>\grave{a}</code>	\dot{a}	<code>\dot{a}</code>	\ddot{a}	<code>\ddot{a}</code>	\breve{a}	<code>\breve{a}</code>
\bar{a}	<code>\bar{a}</code>	\vec{a}	<code>\vec{a}</code>	\widehat{A}	<code>\widehat{A}</code>	\widetilde{A}	<code>\widetilde{A}</code>

표 3.2: 그리스 소문자

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	v	<code>\upsilon</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		

표 3.3: 그리스 대문자

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

⁷이 표들은 David Carlisle 씨가 처음 작성하고 그후 특히 Josef Tkadlec 씨의 제안으로 크게 확장된 `symbols.tex`에서 가져온 것이다.

표 3.4: 이항 관계 연산자

다음 기호들은 그 앞에 `\not` 명령을 붙여서 부정형 기호를 만들 수 있다.

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> or <code>\le</code>	\geq	<code>\geq</code> or <code>\ge</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset^a	<code>\sqsubset^a</code>	\sqsupset^a	<code>\sqsupset^a</code>	\Join^a	<code>\Join^a</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code> , <code>\owns</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
$ $	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\smile	<code>\smile</code>	\frown	<code>\frown</code>	\asymp	<code>\asymp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq	<code>\neq</code> or <code>\ne</code>

^a이 기호를 표시하려면 `latexsym` 패키지가 필요하다.

표 3.5: 이항 연산자

$+$	<code>+</code>	$-$	<code>-</code>	\triangleleft	<code>\triangleleft</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>	\triangleright	<code>\triangleright</code>
\cdot	<code>\cdot</code>	\div	<code>\div</code>	\star	<code>\star</code>
\times	<code>\times</code>	\setminus	<code>\setminus</code>	\ast	<code>\ast</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>	\circ	<code>\circ</code>
\sqcup	<code>\sqcup</code>	\sqcap	<code>\sqcap</code>	\bullet	<code>\bullet</code>
\vee	<code>\vee</code> , <code>\lor</code>	\wedge	<code>\wedge</code> , <code>\land</code>	\diamond	<code>\diamond</code>
\oplus	<code>\oplus</code>	\ominus	<code>\ominus</code>	\uplus	<code>\uplus</code>
\odot	<code>\odot</code>	\oslash	<code>\oslash</code>	\amalg	<code>\amalg</code>
\otimes	<code>\otimes</code>	\bigcirc	<code>\bigcirc</code>	\dagger	<code>\dagger</code>
\triangle	<code>\bigtriangleup</code>	∇	<code>\bigtriangledown</code>	\ddagger	<code>\ddagger</code>
\triangleleft^a	<code>\lhd^a</code>	\triangleright^a	<code>\rhd^a</code>	\wr	<code>\wr</code>
\trianglelefteq^a	<code>\unlhd^a</code>	\trianglerighteq^a	<code>\unrhd^a</code>		

표 3.6: 큰 연산자

\sum	<code>\sum</code>	\bigcup	<code>\bigcup</code>	\bigvee	<code>\bigvee</code>	\bigoplus	<code>\bigoplus</code>
\prod	<code>\prod</code>	\bigcap	<code>\bigcap</code>	\bigwedge	<code>\bigwedge</code>	\bigotimes	<code>\bigotimes</code>
\coprod	<code>\coprod</code>	\bigsqcup	<code>\bigsqcup</code>			\bigodot	<code>\bigodot</code>
\int	<code>\int</code>	\oint	<code>\oint</code>			\biguplus	<code>\biguplus</code>

표 3.7: 화살표

\leftarrow	<code>\leftarrow</code> or <code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\rightarrow	<code>\rightarrow</code> or <code>\to</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftarrow	<code>\Leftarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Longrightarrow	<code>\Longrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Longleftrightarrow	<code>\Longleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>	\iff (bigger spaces)	<code>\iff</code> (bigger spaces)	\leadsto	<code>\leadsto</code> ^a

^a이 기호를 표시하려면 `latexsym` 패키지가 필요하다.

표 3.8: 짝맞춤 기호

$($	<code>(</code>	$)$	<code>)</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
$[$	<code>[</code> or <code>\lbrack</code>	$]$	<code>] or \rbrack</code>	\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>
$\{$	<code>\{ or \lbrace</code>	$\}$	<code>\} or \rbrace</code>	\updownarrow	<code>\updownarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\langle	<code>\langle</code>	\rangle	<code>\rangle</code>	$ $	<code> or \vert</code>	$\ $	<code>\ or \Vert</code>
\lfloor	<code>\lfloor</code>	\rfloor	<code>\rfloor</code>	\lceil	<code>\lceil</code>	\rceil	<code>\rceil</code>
$/$	<code>/</code>	\backslash	<code>\backslash</code>	$.$	<code>(dual. empty)</code>		

표 3.9: 큰 짝맞춤 기호

$\left($	<code>\lgroup</code>	$\right)$	<code>\rgroup</code>	\int	<code>\lmoustache</code>	\int	<code>\rmoustache</code>
\uparrow	<code>\arrowvert</code>	\uparrow	<code>\Arrowvert</code>	$\{$	<code>\bracevert</code>	$\}$	<code>\bracevert</code>

표 3.10: 기타 기호

\dots	<code>\dots</code>	\cdots	<code>\cdots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>
\hbar	<code>\hbar</code>	\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	\aleph	<code>\aleph</code>	\wp	<code>\wp</code>
\forall	<code>\forall</code>	\exists	<code>\exists</code>	\mho^a	<code>\mho^a</code>	∂	<code>\partial</code>
$'$	<code>'</code>	$'$	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∞	<code>\infty</code>
∇	<code>\nabla</code>	\triangle	<code>\triangle</code>	\Box^a	<code>\Box^a</code>	\diamond	<code>\Diamond^a</code>
\bot	<code>\bot</code>	\top	<code>\top</code>	\angle	<code>\angle</code>	\surd	<code>\surd</code>
\diamondsuit	<code>\diamondsuit</code>	\heartsuit	<code>\heartsuit</code>	\clubsuit	<code>\clubsuit</code>	\spadesuit	<code>\spadesuit</code>
\neg or \nmid	<code>\neg</code> or <code>\nmid</code>	\flat	<code>\flat</code>	\natural	<code>\natural</code>	\sharp	<code>\sharp</code>

^a이 기호를 표시하려면 `latexsym` 패키지가 필요하다.

표 3.11: 수학 기호가 아닌 것

이 기호들은 텍스트 모드에서도 사용가능하다.

\dagger	<code>\dag</code>	\S	<code>\S</code>	\copyright	<code>\copyright</code>	\textregistered	<code>\textregistered</code>
\ddagger	<code>\ddag</code>	\P	<code>\P</code>	\pounds	<code>\pounds</code>	$\%$	<code>\%</code>

표 3.12: AMS 짝맞춤 기호

\ulcorner	<code>\ulcorner</code>	\urcorner	<code>\urcorner</code>	\llcorner	<code>\llcorner</code>	\lrcorner	<code>\lrcorner</code>
\lvert	<code>\lvert</code>	\rvert	<code>\rvert</code>	\lVert	<code>\lVert</code>	\rVert	<code>\rVert</code>

표 3.13: AMS 그리스 및 헤브루 문자

\digamma	<code>\digamma</code>	\varkappa	<code>\varkappa</code>	\beth	<code>\beth</code>	\gimel	<code>\gimel</code>	\daleth	<code>\daleth</code>
------------	-----------------------	-------------	------------------------	---------	--------------------	----------	---------------------	-----------	----------------------

표 3.14: AMS 이항 관계 연산자

\lessdot	<code>\lessdot</code>	\gtrdot	<code>\gtrdot</code>	\doteqdot or \Doteq	<code>\doteqdot</code> or <code>\Doteq</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>	\risingdotseq	<code>\risingdotseq</code>
\eqslantless	<code>\eqslantless</code>	\eqslantgtr	<code>\eqslantgtr</code>	\fallingdotseq	<code>\fallingdotseq</code>
\leqq	<code>\leqq</code>	\geqq	<code>\geqq</code>	\eqcirc	<code>\eqcirc</code>
\lll or \llless	<code>\lll</code> or <code>\llless</code>	\ggg or \gggtr	<code>\ggg</code> or <code>\gggtr</code>	\circeq	<code>\circeq</code>
\lesssim	<code>\lesssim</code>	\gtrsim	<code>\gtrsim</code>	\triangleq	<code>\triangleq</code>
\lessapprox	<code>\lessapprox</code>	\gtrapprox	<code>\gtrapprox</code>	\bumpeq	<code>\bumpeq</code>
\lessgtr	<code>\lessgtr</code>	\gtrless	<code>\gtrless</code>	\Bumpeq	<code>\Bumpeq</code>
\lesseqgtr	<code>\lesseqgtr</code>	\gtreqless	<code>\gtreqless</code>	\thicksim	<code>\thicksim</code>
\lesseqqgtr	<code>\lesseqqgtr</code>	\gtreqqless	<code>\gtreqqless</code>	\thickapprox	<code>\thickapprox</code>
\preccurlyeq	<code>\preccurlyeq</code>	\succcurlyeq	<code>\succcurlyeq</code>	\approxeq	<code>\approxeq</code>
\curlyeqprec	<code>\curlyeqprec</code>	\curlyeqsucc	<code>\curlyeqsucc</code>	\backsim	<code>\backsim</code>
\precsim	<code>\precsim</code>	\succsim	<code>\succsim</code>	\backsimeq	<code>\backsimeq</code>
\precapprox	<code>\precapprox</code>	\succapprox	<code>\succapprox</code>	\vDash	<code>\vDash</code>
\subseteqq	<code>\subseteqq</code>	\supseteqq	<code>\supseteqq</code>	\Vdash	<code>\Vdash</code>
\Subset	<code>\Subset</code>	\Supset	<code>\Supset</code>	\Vvdash	<code>\Vvdash</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\backepsilon	<code>\backepsilon</code>
\therefore	<code>\therefore</code>	\because	<code>\because</code>	\varpropto	<code>\varpropto</code>
\shortmid	<code>\shortmid</code>	\shortparallel	<code>\shortparallel</code>	\between	<code>\between</code>
\smallsmile	<code>\smallsmile</code>	\smallfrown	<code>\smallfrown</code>	\pitchfork	<code>\pitchfork</code>
\vartriangleleft	<code>\vartriangleleft</code>	\vartriangleright	<code>\vartriangleright</code>	\blacktriangleleft	<code>\blacktriangleleft</code>
\trianglelefteq	<code>\trianglelefteq</code>	\trianglerighteq	<code>\trianglerighteq</code>	\blacktriangleright	<code>\blacktriangleright</code>

표 3.15: AMS 화살표

\dashleftarrow	<code>\dashleftarrow</code>	\dashrightarrow	<code>\dashrightarrow</code>	\multimap	<code>\multimap</code>
\leftrightsquigarrow	<code>\leftrightsquigarrow</code>	\rightleftarrows	<code>\rightleftarrows</code>	\upuparrows	<code>\upuparrows</code>
\leftrightsquigarrow	<code>\leftrightsquigarrow</code>	\rightleftarrows	<code>\rightleftarrows</code>	\downdownarrows	<code>\downdownarrows</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Rrightarrow	<code>\Rrightarrow</code>	\upharpoonleft	<code>\upharpoonleft</code>
\twoheadleftarrow	<code>\twoheadleftarrow</code>	\twoheadrightarrow	<code>\twoheadrightarrow</code>	\upharpoonright	<code>\upharpoonright</code>
\leftarrowtail	<code>\leftarrowtail</code>	\rightarrowtail	<code>\rightarrowtail</code>	\downharpoonleft	<code>\downharpoonleft</code>
\leftrightharpoons	<code>\leftrightharpoons</code>	\rightleftharpoons	<code>\rightleftharpoons</code>	\downharpoonright	<code>\downharpoonright</code>
\Lsh	<code>\Lsh</code>	\Rsh	<code>\Rsh</code>	\rightsquigarrow	<code>\rightsquigarrow</code>
\looparrowleft	<code>\looparrowleft</code>	\looparrowright	<code>\looparrowright</code>	\leftrightsquigarrow	<code>\leftrightsquigarrow</code>
\curvearrowleft	<code>\curvearrowleft</code>	\curvearrowright	<code>\curvearrowright</code>		
\circlearrowleft	<code>\circlearrowleft</code>	\circlearrowright	<code>\circlearrowright</code>		

표 3.16: AMS 부정 이항 관계 연산자 및 화살표

\nless	\ngtr	\varsubsetneqq
\lneq	\gneq	\varsupsetneqq
\nleq	\ngeq	\nsubseteqeq
\nleqslant	\ngeqslant	\nsupseteqeq
\lneqq	\gneqq	\nmid
\lvertneqq	\gvertneqq	\nparallel
\nleqq	\ngeqq	\nshortmid
\lnsim	\gnsim	\nshortparallel
\lnapprox	\gnapprox	\nsim
\nprec	\nsucc	\ncong
\npreceq	\nsucceq	\nvdash
\precneqq	\succneqq	\nvDash
\precnsim	\succnsim	\nVdash
\precnapprox	\succnapprox	\nVDash
\subsetneq	\supsetneq	\ntriangleleft
\varsubsetneq	\varsupsetneq	\ntriangleright
\nsubseteq	\nsupseteq	\ntrianglelefteq
\subsetneqq	\supsetneqq	\ntrianglerighteq
\nleftarrow	\rightarrow	\nleftrightarrow
\nLeftarrow	\nrightarrow	\nLeftrightarrow

표 3.17: AMS 이항 연산자

$\dot{+}$	\centerdot	\intercal
\ltimes	\rtimes	\divideontimes
\Cup or \doublecup	\Cap or \doublecap	\smallsetminus
\veebar	\barwedge	\doublebarwedge
\boxplus	\boxminus	\circledast
\boxtimes	\boxdot	\circledcirc
\leftthreetimes	\rightthreetimes	\circledR
\curlyvee	\curlywedge	

표 3.18: AMS 기타 기호

\hbar	<code>\hbar</code>	\hslash	<code>\hslash</code>	\Bbbk	<code>\Bbbk</code>
\square	<code>\square</code>	\blacksquare	<code>\blacksquare</code>	\textcircled{S}	<code>\circledS</code>
\triangle	<code>\vartriangle</code>	\blacktriangle	<code>\blacktriangle</code>	\complement	<code>\complement</code>
∇	<code>\triangledown</code>	\blacktriangledown	<code>\blacktriangledown</code>	\Game	<code>\Game</code>
\lozenge	<code>\lozenge</code>	\blacklozenge	<code>\blacklozenge</code>	\bigstar	<code>\bigstar</code>
\angle	<code>\angle</code>	\measuredangle	<code>\measuredangle</code>	\sphericalangle	<code>\sphericalangle</code>
\diagup	<code>\diagup</code>	\diagdown	<code>\diagdown</code>	\backprime	<code>\backprime</code>
\nexists	<code>\nexists</code>	\Finv	<code>\Finv</code>	\varnothing	<code>\varnothing</code>
\eth	<code>\eth</code>	\mho	<code>\mho</code>		

표 3.19: 수학 알파벳

보기	명령	필요한 패키지
$\mathrm{ABCDEabcde1234}$	<code>\mathrm{ABCDE abcde 1234}</code>	
$\mathit{ABCDEabcde1234}$	<code>\mathit{ABCDE abcde 1234}</code>	
$\mathnormal{ABCDEabcde1234}$	<code>\mathnormal{ABCDE abcde 1234}</code>	
\mathcal{ABCDE}	<code>\mathcal{ABCDE abcde 1234}</code>	
\mathscr{ABCDE}	<code>\mathscr{ABCDE abcde 1234}</code>	<code>mathrsfs</code>
$\mathfrak{ABCDEabcde1234}$	<code>\mathfrak{ABCDE abcde 1234}</code>	<code>amsfonts</code> or <code>amssymb</code>
$\mathbb{ABCDE\Im\Re\X\Y\Z}$	<code>\mathbb{ABCDE abcde 1234}</code>	<code>amsfonts</code> or <code>amssymb</code>

