

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

Chapter 4~5

특별한 기능

수학 관련 그림 그리기

KINS

2013년 1월 1일

제 4장

특별한 기능

외부 그림 넣기

외부 그림을 처리하기 위해서는 `graphicx`패키지를 사용한다. 즉,

```
\usepackage[pdftex]{color, graphicx}
```

명령을 전처리부에 넣으면 된다.

외부 그림 넣기

패키지를 로드한 후, 그림을 넣기 원하는 곳에 다음 명령어를 쓴다.

```
\includegraphics[ key=value, ... ]{ file name }
```

이 때, key인자는 다음과 같다.

key인자	설명
width=#	삽입될 그림의 폭을 # 으로 지정. 그림의 높이 (height)를 명시하지 않으면 가로:세로 비율을 유지함.
height=#	삽입될 그림의 높이를 # 으로 지정. 그림의 폭 (width)를 명시하지 않으면 가로:세로 비율을 유지함.
angle=#	그림의 각도를 # 만큼 반시계 방향으로 회전. 단위는 도(°).
scale=#	그림을 전체적으로 # 배 늘이거나 줄임.

외부 그림 넣기

```
\includegraphics[ key=value, ... ]{ file name }
```

- ▶ pdf \LaTeX 에서 처리할 수 있는 그림은 .pdf, .jpg, .png 형식이다.
- ▶ file name은 넣고자 하는 그림의 파일명이며 파일은 현재 작업중인 폴더에 있어야 한다.
- ▶ 그림을 떠다니는 객체로 만들고자 한다면 위 명령을 `\begin{figure}`와 `\end{figure}`사이에 위치시킨다.

외부 그림 넣기

```
\includegraphics[width=0.2\textwidth]{ein.jpg}
```

```
\includegraphics[width=0.2\textwidth, height=0.1\textwidth]{ein.jpg}
```

```
\includegraphics[scale=0.1]{ein.jpg}
```

```
\includegraphics[scale=0.2]{ein.jpg}
```

```
\includegraphics[scale=0.2, angle=45]{ein.jpg}
```



제 5장

수학 관련 그림 그리기

picture환경

picture환경은 \LaTeX 에 내장된 환경이고, 텍 프로그램 안에서 그림을 그리게 해 준다.

따라서 picture환경으로 그린 그림은 텍으로 작성한 **문서의 다른 부분과 완벽하게 호환**된다.

picture환경

picture환경은 다음 두 가지 형식으로 선언한다:

```
\begin{picture}(x,y)           \begin{picture}(x,y)(x',y')
...                           또는, ...
\end{picture}                 \end{picture}
```

- ▶ x, y 인자는 그림이 그려질 영역의 크기를 지정하는데 단위는 `\unitlength` (기본값 1pt) 이다.
- ▶ x', y' 인자는 영역의 왼쪽 아래의 좌표값을 설정해준다.
- ▶ `\unitlength`는 picture환경 밖에서 `\setlength{\unitlength}{ length }` 명령을 쓰면 언제든지 바꿀 수 있다.

picture환경

picture환경 내에서 어떤 위치에 object를 두려면 다음 명령을 쓴다:

```
\put(x,y){ object }      또는, \put(x,y)(x',y'){ n }{ object }
```

object는 보통 `\line(x,y){ length }`, `\vector(x,y){ length }` 등의 코드나 문자열이 된다.

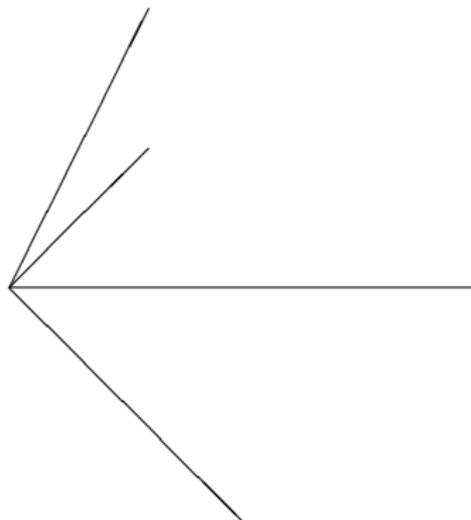
picture환경 - 선분 그리기

선분은 `\put(a,b){\line(x,y){ length }}` 명령을 사용한다.

- ▶ (a, b) 인자는 선분의 시작점의 좌표를 나타낸다.
- ▶ (x, y) 인자는 선분의 방향벡터를 나타내는데, x 와 y 는 서로소이며 각각 -6 과 6 사이의 값을 가진다.
- ▶ `length` 인자는 **수평 또는 수직방향길이**를 나타낸다. `\unitlength`에 비례한다.

예시

```
\setlength{\unitlength}{\textwidth}  
\begin{picture}(1,1)  
\put(0,0.5){\line(1,0){1}}  
\put(0,0.5){\line(1,1){0.3}}  
\put(0,0.5){\line(1,2){0.3}}  
\put(0,0.5){\line(1,-1){0.5}}  
\end{picture}
```



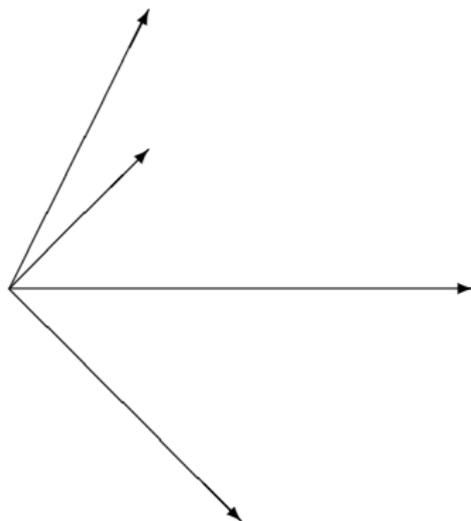
picture환경 - 화살표 또는 벡터 그리기

벡터는 `\put(a,b){\vector(x,y){ length }}` 명령을 사용한다.

- ▶ (a, b) 인자는 벡터의 시작점의 좌표를 나타낸다.
- ▶ (x, y) 인자는 벡터의 방향벡터를 나타내는데, x 와 y 는 서로소이며 각각 -4 과 4 사이의 값을 가진다.
- ▶ `length` 인자는 **수평 또는 수직방향길이**를 나타낸다. `\unitlength`에 비례한다.

예시

```
\setlength{\unitlength}{\textwidth}  
\begin{picture}(1,1)  
\put(0,0.5){\vector(1,0){1}}  
\put(0,0.5){\vector(1,1){0.3}}  
\put(0,0.5){\vector(1,2){0.3}}  
\put(0,0.5){\vector(1,-1){0.5}}  
\end{picture}
```



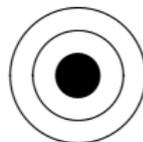
picture환경 - 원 그리기

원은 `\put(a,b){\circle{ length }}` 명령을 사용한다.

- ▶ (a, b) 인자는 원의 중심의 좌표를 나타낸다.
- ▶ `length` 인자는 원의 지름을 나타낸다. 단위는 `\unitlength`이다.
- ▶ `\put(a,b){\circle*{ length }}` 명령은 내부가 짝 찬 원을 그린다.

예시

```
\setlength{\unitlength}{\textwidth}  
\begin{picture}(1,1)  
\put(.5,.5){\circle{0.2}}  
\put(.5,.5){\circle{0.3}}  
\put(.5,.5){\circle*{0.1}}  
\put(.1,.1){\circle{0.1}}  
\end{picture}
```



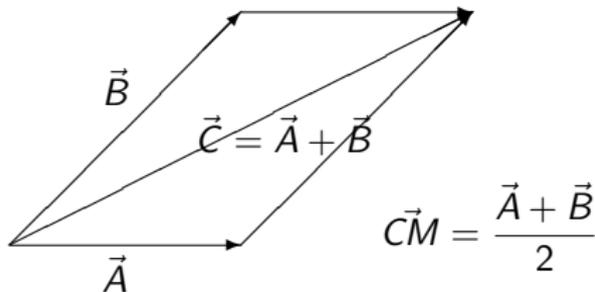
picture환경 - 텍스트와 수식

텍스트와 수식은 `\put(a,b){ text }` 명령을 사용한다.

- ▶ (a, b) 인자는 텍스트가 입력될 위치를 나타낸다.
- ▶ `\put` 명령 안에서는 일반적인 텍스트, 수식 입력 방식에 따라 입력하면 된다.
- ▶ `\put(a,b){ $\displaystyle\dots$ }` 명령은 입력된 수식을 display 스타일로 보여준다.

예시

```
\setlength{\unitlength}{\textwidth}
\begin{picture}(1,1)
\put(0,.2){\vector(1,0){0.5}}
\put(0,.2){\vector(1,1){0.5}}
\put(0.5,.2){\vector(1,1){0.5}}
\put(.5,.7){\vector(1,0){0.5}}
\put(0,.2){\vector(2,1){1}}
\put(.2,.1){\vec{A}}
\put(.2,.5){\vec{B}}
\put(.4,.4){\vec{C}=\vec{A}+\vec{B}}
\put(.8,.2){\displaystyle \vec{CM}=\frac{\vec{A}+\vec{B}}{2}}
\end{picture}
```



picture환경 - 알 모양 곡선 그리기

알 모양 곡선은 `\put(a,b){\oval(w,h)[part]}` 명령을 사용한다.

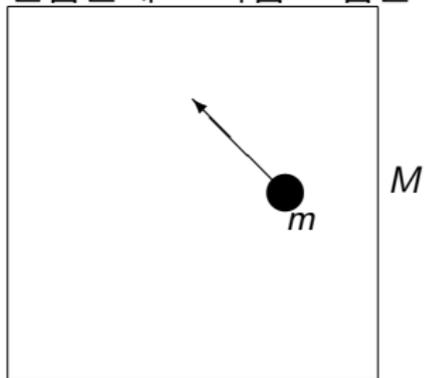
- ▶ (a, b) 인자는 곡선의 중심이다.
- ▶ (w, h) 인자는 각각 곡선의 장축과 단축을 지정한다.
- ▶ `part` 인자는 곡선이 그려질 부분을 지정하며 t(위), b(아래), l(왼쪽), r(오른쪽)을 인자로 취한다.

예시

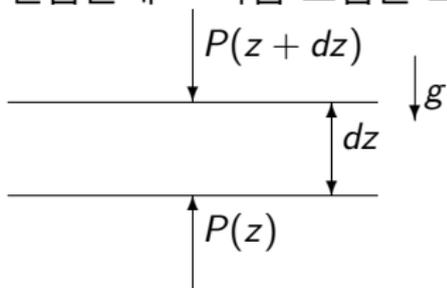
```
\setlength{\unitlength}{\textwidth}  
\begin{picture}(1,1)  
\put(.5,.5){\oval(.1,.1)}  
\put(.5,.5){\oval(.2,.1)}  
\put(.5,.5){\oval(.3,.2)[t]}  
\put(.1,.1){\oval(.2,.1)[tr]}  
\end{picture}
```



- ▶ 연습문제 1. 다음 그림을 그려라.



- ▶ 연습문제 2. 다음 그림을 그려라.



TikZ환경

picture환경보다 직관적이고 강력한 그림 그리기 환경이 바로 TikZ이다. 이는 다음과 같이 package형태로 불러들여야 한다.

```
\usepackage{tikz}
```

또한 TikZ 그림은 다음과 같은 환경

```
\begin{tikzpicture}  
...contents...  
\end{tikzpicture}
```

또는 명령어

```
\tikz{ ...contents... }
```

안에서 이루어진다.

TikZ환경-기본 사항

- ▶ TikZ의 기본 단위는 cm이다. 단위 없이 쓰인 모든 양은 cm를 단위로 취한다.
- ▶ TikZ의 기본 명령은 `\draw[option] object ;` 이다. 이 명령이 시전되면 \LaTeX 은 투명한 **도화지 한 장(Layer)**과 option이라는 속성을 가진 **연필 한 자루**를 받게 된다. 그리고 draw명령과 세미콜론(;)사이에 오는 object의 지시에 따라 그림을 그리게 된다. **세미콜론을 빼 먹으면 곤란하다.**
- ▶ object는 점(point)과 경로(path)로 이루어져 있는데, 여기서 경로란 **연필의 움직임을 \LaTeX 에게 알려주는 명령**이라고 생각하면 된다.

TikZ환경-기본 사항

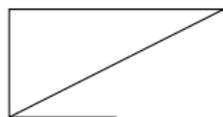
- ▶ 점을 지정하는 방법은 크게 두 가지가 있다:
 - ▶ 직교좌표계 형태로 지정하기: (a,b) .
 - ▶ 극좌표계 형태로 지정하기: $(\theta: r)$.
 - ▶ `\path (a,b) coordinate (P)` 명령어를 사용해 (P)에 특정 좌표 (a,b) 를 지정하기. 이렇게 지정하면, (a,b) 자리에 (P)를 쓸 수 있다.

이 때, a,b 는 단위를 명시하지 않아도 되지만 r 은 **단위를 명시해야 한다**. TikZ은 **상대적 좌표를 사용**하므로 원점을 어떻게 잡느냐는 중요하지 않다.

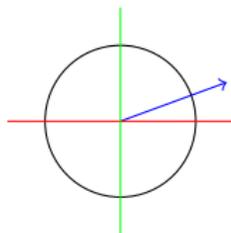
- ▶ 경로(path)는 연필이 현재 위치에서 어떻게 움직여야 하며 최종적으로 어느 위치에 멈춰야 하는지에 대한 정보를 모두 가지고 있다.

TikZ환경-가장 간단한 예제

```
\begin{tikzpicture}
\draw (0,0) -- (1,0) ;
\draw (0,0) -- (2,1)
-- (0,1)--(0,0);
\end{tikzpicture}
```



```
\begin{tikzpicture}
\draw (0,0) circle (2em) ;
\draw[->,color=blue] (0,0)
-- (20:3em);
\draw[color=red] (-3em,0)
-- (3em,0);
\draw[color=green] (0,-3em)
-- (0,3em);
\end{tikzpicture}
```



TikZ환경

TikZ으로 무엇을 그릴 수 있는가? 우선 기본 도형으로는..

- ▶ **(a,b)** : 아무것도 그리지 않으며, 연필을 (a,b)위치에 둔다.
- ▶ **initial - - final**: initial 과 final을 잇는 선분을 그린 후, 연필을 final 위치에 둬.
- ▶ **ld rectangle ru** : ld와 ru을 양 끝 대각선으로 하는 직사각형을 그린 후, 연필을 ru위치에 둬.
- ▶ **c circle (r)**: 중심이 c이고 반지름이 r인 원을 그린 후 연필을 c 위치에둬. **r은 단위를 명시해야 함.**
- ▶ **c arc ($\theta_1 : \theta_2 : r$)**: 중심이 c이고 반지름이 r이며 양 끝 각도가 θ_1 , θ_2 인 호를 그린 후 연필을 c위치에 둬. **r은 단위를 명시해야 함.**
- ▶ **c ellipse (r1 and r2)** 두 축의 길이가 r1, r2이고 중심이 c인 타원을 그린 후 연필을 c위치에 둬. **r1, r2는 단위를 명시해야 함.**

TikZ환경

좀 더 복잡한 도형으로는...

- ▶ **plot($\backslash x, f(\backslash x)$)** : $f(x)$ 의 그래프를 찍어준다. `\draw`명령의 option 인자로 정의역을 `domain=a:b` 형태로 명시해야 한다.
- ▶ **node[*position*]{}**: 현재 연필이 놓인 자리를 기준으로 *position* 위치에 {}안의 객체를 식자한다. *position*인자로는 `right`, `left`, `above`, `below`, `below(above) left`, `below(above) right` 가 쓰인다. 이를 명시하지 않으면 현재 연필의 위치 위에 식자한다.
- ▶ **ld grid ru** ld과 ru을 양 끝 대각선으로 하는 직사각형 격자를 그림. `draw`의 옵션인자 `step=#` 으로 격자 사이의 간격을 지정할 수 있다.

TikZ환경

다음은 draw[option] 에서 option인자로 올 수 있는 것들이다.

option	설명
color=#	선의 색을 지정. # 에는 red, blue, gray, green, orange, cyan, brown, lime 등 xcolor 패키지에서 미리 정의된 색상 이름이 온다.
->, ->>, <->	각각 한쪽 화살표, 한쪽 이중 화살표, 양 끝 화살표가 있는 경로를 그린다.
line width=#	선의 굵기를 결정한다. line width=# 대신 thin, very thin, ultra thin, thick, very thick, ultra thick 등의 인자를 써도 된다.

TikZ환경

option	설명
dash patter=#	점선 형태를 결정하는데, - off - on - off -on .. 형태로 입력한다. 예컨대, 1pt off 2pt on 3pt off 2pt로 쓰면, 1pt짜리 선을 긋고, 2pt를 비우고 다시 3pt 선을 긋고 2pt를 비우는 것을 반복하라는 뜻이다.
dotted	점선을 긋는다.
dashed	대쉬 라인으로 선을 긋는다.
>=#	화살표의 팁 모양을 # 으로 바꾼다. # 에 latex 이라는 글자를 쓰면 \LaTeX 표준 화살표 팁을 사용할 수 있다.

TikZ환경

- ▶ 어떤 도형의 내부를 채우려면 `\draw` 대신 `\fill[option]` ; 명령어를 쓰면 된다. 이 명령의 사용법은 `draw`와 동일하고, `option` 인자에 들어가는 색상은 채우기 색상으로 처리된다.
- ▶ 도형의 외곽선 색과 채움 색을 다르게 하려면 `\filldraw[fill=#1, draw=#2,...]` ;명령을 쓴다. 이 때, # 1 은 채우기 색, # 2는 외곽선 색을 나타낸다.
- ▶ 채우기를 하기 위해서는 매끄러운 닫힌 곡선이 필요한데, 이는 `object` 마지막에 `-- cycle` 을 추가하면 된다.

TikZ환경

어떤 도형을 특정 규칙에 따라 반복해서 그리고자 한다면, draw(fill, filldraw)명령 앞에

```
\foreach \t in { variable }
```

를 추가하면 된다. 이 때, draw 명령 안에 있는 \t 가 각각의 variable로 치환되어 수행된다. t는 임의의 다른 문자를 써도 되고, variable은 t가 취할 수 있는 것(숫자, 문자, etc)을 나타낸다. 원소나열법으로 써도 좋고, 정수인 경우 {a,...,b} 이라고 쓰면 a,b사이의 정수로 이해한다.

TikZ환경

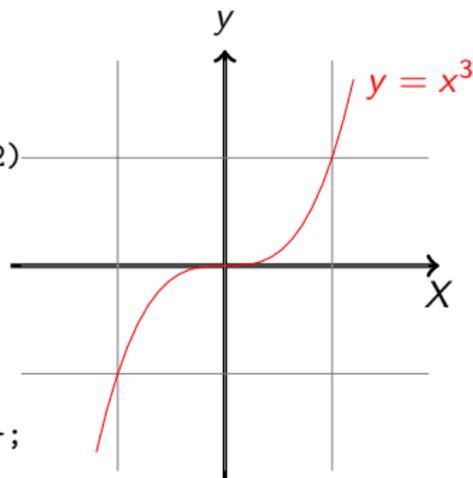
tikzpicture환경 내에서 다음 scope환경을 사용하면 환경 안에서 특정 옵션을 어떤 값으로 유지할 수 있다.

```
\begin{scope}[option=#]  
...  
\draw[->] ....;  
...  
\end{scope}
```

이 때, scope환경 안에 있는 모든 option은 따로 지정하지 않아도 #이라는 값을 가지게 된다.

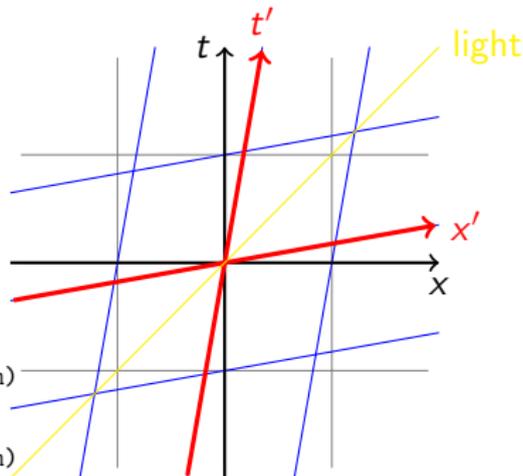
TikZ환경

```
\begin{tikzpicture}
\draw[->, very thick]
(-2, 0)--(2,0) node[below]{$X$};
\draw[->, very thick] (0, -2)--(0,2)
node[above]{$y$};
\draw[color=gray, step=1]
(-1.9,-1.9) grid (1.9, 1.9);
\draw[color=red, domain=-1.2:1.2]
plot(\x,\x^3) node[right] {$y=x^3$};
\end{tikzpicture}
```

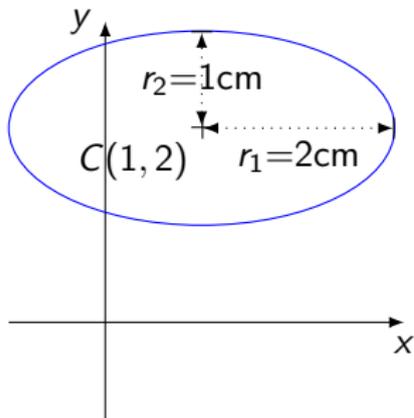


TikZ환경

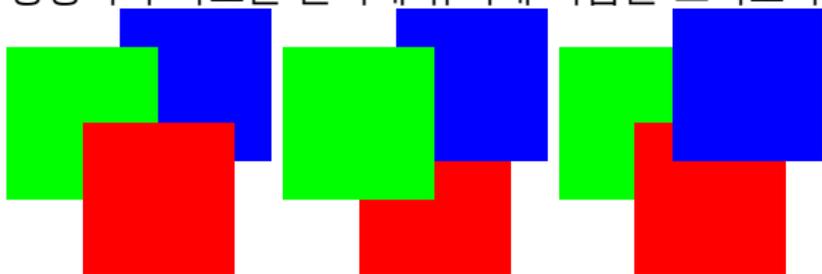
```
\begin{tikzpicture}
\draw[thin, color=gray, step=1cm]
(-1.9,-1.9) grid (1.9,1.9);
\foreach \t in {-1,...,1}
\draw[color=blue] (-2, -0.176*2+\t)
--(2, 0.176*2+\t);
\foreach \y in {-1,0,1}
\draw[color=blue] (-0.176*2+\y,-2)
--(0.176*2+\y,2);
\draw[->, thick] (-2,0)--(2,0)
node[below] {$x$};
\draw[->, thick] (0,-2)--(0,2)
node[left] {$t$};
\draw[->, color=red, very thick] (10:-2cm)
--(10:2cm) node[right] {$x'$};
\draw[->, color=red, very thick] (80:-2cm)
--(80:2cm) node[above] {$t'$};
\draw[color=yellow] (-2,-2)--(2,2)
node[right] {light};
\end{tikzpicture}
```



- ▶ 연습문제 3. 다음을 그려라.



- ▶ 연습문제 4. draw 명령어가 나오는 순서가 중요할까? draw 명령어가 나오는 순서에 유의해 다음을 그려보라.



ipe의 사용

이 모든 작업을 \LaTeX 안에서 하기엔 너무 힘들다...
따라서, 비록 완벽한 호환성을 얻진 못하지만 차선책으로 ipe라는
툴을 쓸 수 있다.

ipe의 사용

ipe의 설치: <http://ipe7.sourceforge.net/>

- ▶ ipe는 vector graphic tool이고 picture환경의 \LaTeX 그림과 결과물의 모양이 매우 흡사하다.
- ▶ \LaTeX 코드를 그림에 직접 삽입할 수 있다.
- ▶ .eps또는 .pdf형식을 지원한다.

Xy-pic의 사용

Xy-pic은 수학의 다이어그램을 쉽게 그릴 수 있게 해 주는 패키지이며, 다음과 같이 로드한다.

```
\usepackage[all]{xy}
```

option 인자로 all을 사용해서 Xy-pic의 모든 명령어를 불러오는 것이 좋다.

Xy-pic의 사용

Xy를 사용한 다이어그램은 기본적으로 Xy-matrix 위에 그려지며 Xy-matrix는 수학 모드에서 다음과 같이 선언한다.

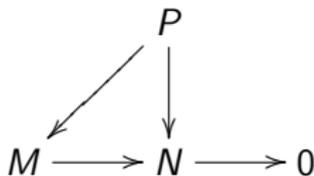
```
\[
\xymatrix{
  & P & & \\
M & N & & O \\
}
\]
```

M *N* 0

Xy-pic의 사용

여기에 화살표를 그리려면 화살표가 시작하는 셀에 `\ar[]` 명령을 쓴다. []안의 인자는 화살표의 방향을 나타내며 u,d,r,l를 쓸 수 있다.

```
\[
\xymatrix{
  & P\ar[d]\ar[dl] & \\
M\ar[r] & N\ar[r] & 0
}
\]
```

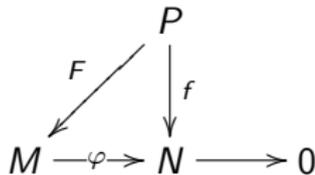


Xy-pic의 사용

때로는 화살표에 글씨나 기호를 넣고 싶을 때가 있다. 이 경우 첨자 명령어 \wedge , $_$ 와 수직선 $|$ 를 사용한다.

위첨자는 화살표 진행방향 왼쪽에, 아래첨자는 화살표 진행방향 오른쪽, 수직선은 화살표 위에 기호를 넣는다.

```
\[
\xymatrix{
  & P \ar[d]^\{f\} \ar[dl]_{F} & \\
M \ar[r] | \{\varphi\} & N \ar[r] & 0
}
\]
```



Xy-pic의 사용

화살표의 모양을 바꾸려면 `\ar` 뒤에 모양을 지정하는 `@{...}`을 사용하면 된다.

```
\[
```

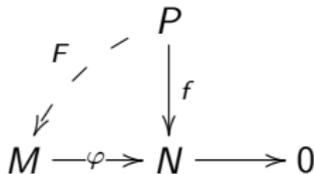
```
\xymatrix{
```

```
& P\ar[d]^{f}\ar@/_/@{-->}[dl]_{F} & \\\
```

```
M\ar[r]|{\varphi} & N\ar[r] & 0
```

```
}
```

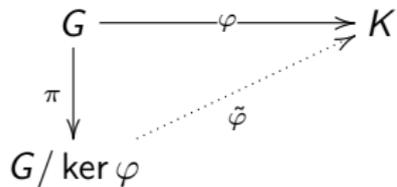
```
\]
```



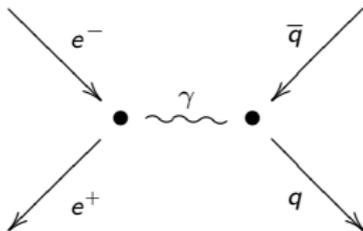
주로 사용되는 모양은 `@{.>}`, `@{~}`, `@{-->}`, `@{^{\{ }->}`, `@{=>}` 등이 있다.

또한, `@/.../`를 사용하면 화살표가 그려지는 모양을 정할 수 있다.

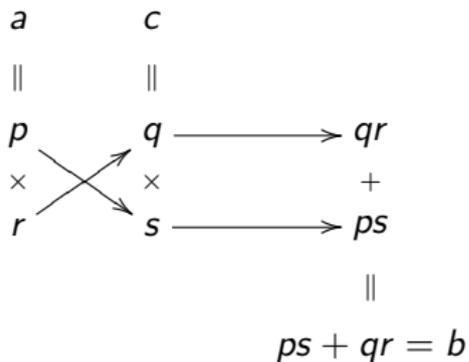
- ▶ 연습문제 5. 다음을 그리시오.



- ▶ 연습문제 6. 다음을 그리시오.



▶ 연습문제 7. 다음을 그리시오.



(**HINT.** $\ar@{\times}[\dots]$ 는 ‘투명한’ 화살표를 그려준다.)