

# picture

L<sup>A</sup>T<sub>E</sub>X Built-in Drawing Environment

## 오늘 알아볼 내용

### picture 환경

L<sup>A</sup>T<sub>E</sub>X에서 기본으로 제공하는 '간단한' 그리기 환경

### epic, eepic, curves 패키지

picture의 한계를 뛰어 넘으려는 노력

### pict2e 패키지

11년만에 완성된 '간단한' L<sup>A</sup>T<sub>E</sub>X드로잉 패키지

### curve2e 패키지

L<sup>A</sup>T<sub>E</sub>X, 곡선에서 놀다

## 최종 목표

다음 그림을 그려본다.

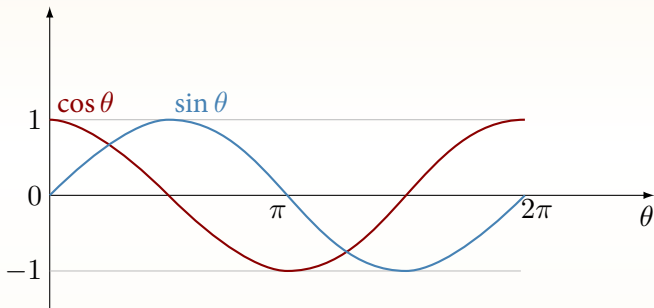


Figure:  $\sin \theta$ 와  $\cos \theta$ 의 그래프 ( $0 \leq \theta \leq 2\pi$ )

## 지금 설명할 것은

### picture 환경

LaTeX에서 기본으로 제공하는 '간단한' 그리기 환경

### epic, eepic, curves 패키지

picture의 한계를 뛰어 넘으려는 노력

### pict2e 패키지

11년만에 완성된 '간단한' LaTeX드로잉 패키지

### curve2e 패키지

LaTeX, 곡선에서 놀다

## Hello, picture?

- $\text{\LaTeX}$ 에 내장된 (built-in) '간단한' 드로잉 환경
- 별도의 그래픽 툴 없이 텍스트를 편집하면서 그림을 그려넣는다.
- 타이포그래피적 관점
  - ⇒ 그림과 본문의 폰트를 일치시켜 완성도 높은 문서 생산 가능

## 그림을 그리기 전에 알아두어야 할 것

`\unitlength` 앞으로 사용할 단위 길이

TeX이 알아듣는 모든 길이 단위 사용 가능

```
\setlength{\unitlength}{1pt} % 기본값
```

```
\setlength{\unitlength}{3mm}
```

```
\setlength{\unitlength}{0.5cm}
```

```
\setlength{\unitlength}{0.1in}
```

```
\setlength{\unitlength}{2pc}
```

...

`\picture` 그리기 시작할 때 필수적인 환경

```
\begin{picture}(가로 폭,세로 폭)(x-좌표 시작점, y-좌표 시작점)
```

...

```
\end{picture}
```

`\put` 원하는 좌표에 그림 요소를 놓을 때

```
\put(x,y){그림 요소}
```

## 간단한 예제

```
\setlength\unitlength{1mm}
```

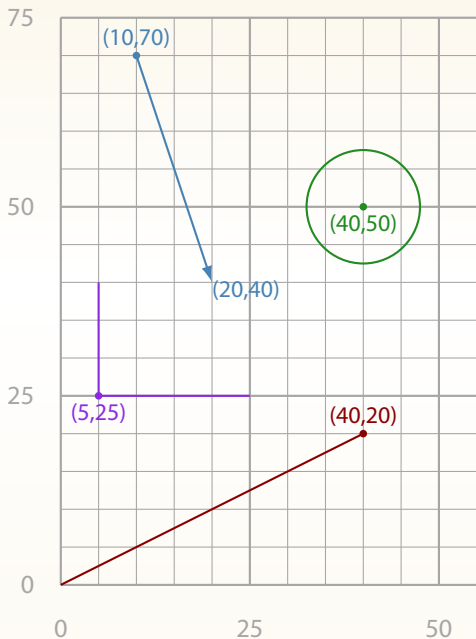
```
\begin{picture}(50,70)
```

```
\put(0,0){\line(2,1){40}}  
기울기가 1/2이고 원점에서  
수평으로 40단위 뺀 직선
```

```
\put(5,25){\line(1,0){20}}  
시점에서 x축에 평행한 20단위 직선  
\put(5,25){\line(0,1){15}}  
시점에서 y축에 평행한 15단위 직선
```

```
\put(10,70){\vector(1,-3){10}}  
기울기가 -3이고 시점에서  
수평방향으로 10단위 뺀 벡터
```

```
\put(40,50){\circle{15}}  
중점 (40,50)에 지름 15단위의 원  
\end{picture}
```

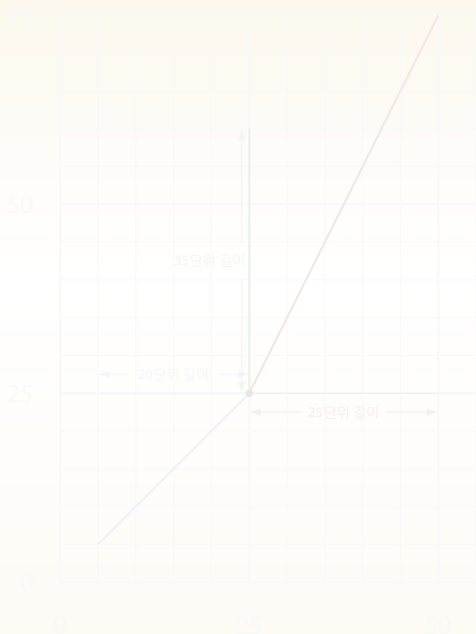


## \line 직선

`\line(x,y){길이}`

- $(x,y)$ 는 기울기
- {길이}의 개념 주의!
  - **y축에 평행한 직선**  
y축 방향으로 주어진 길이  
단위 만큼 이동
  - **그렇지 않은 직선**  
시점에서 x축 방향으로  
주어진 길이 단위 만큼 이동

```
\put(25,25){\line(1,2){25}}  
\put(25,25){\line(-1,-1){20}}  
\put(25,25){\line(0,1){35}}
```



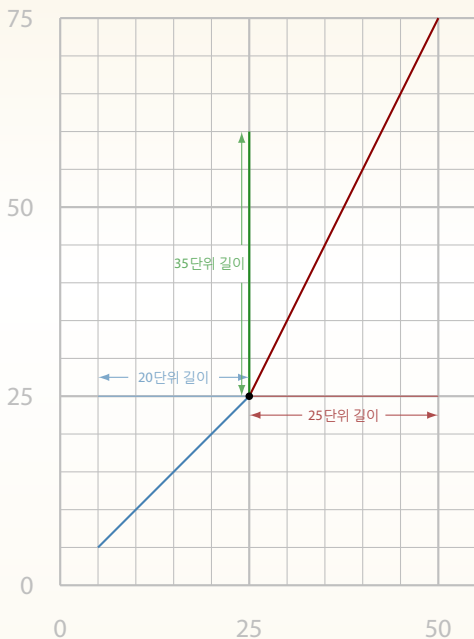


## `\line` 직선

`\line(x,y){길이}`

- $(x,y)$ 는 기울기
- {길이}의 개념 주의!
  - **y축에 평행한 직선**  
y축 방향으로 주어진 길이  
단위 만큼 이동
  - **그렇지 않은 직선**  
시점에서 x축 방향으로  
주어진 길이 단위 만큼 이동

```
\put(25,25){\line(1,2){25}}  
\put(25,25){\line(-1,-1){20}}  
\put(25,25){\line(0,1){35}}
```

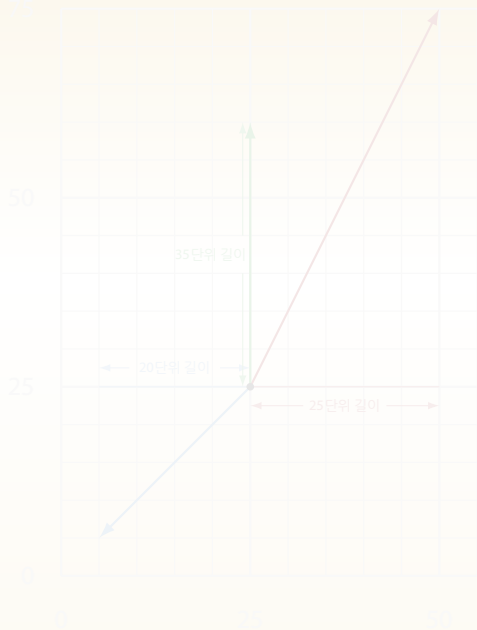


# \vector 벡터

`\vector(x,y){길이}`

- $(x,y)$ 는 기울기
- {길이}의 개념은 `\line`과 같음
- 끝점에 화살촉

```
\put(25,25){\vector(1,2){25}}  
\put(25,25){\vector(-1,-1){20}}  
\put(25,25){\vector(0,1){35}}
```

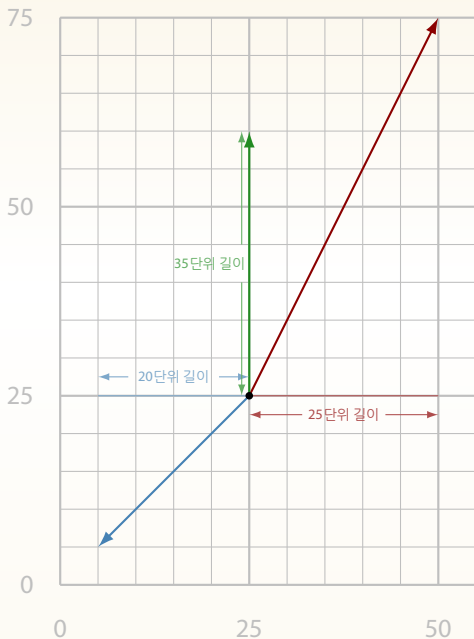


# \vector 벡터

`\vector(x,y){길이}`

- $(x, y)$ 는 기울기
- {길이}의 개념은 `\line`과 같음
- 끝점에 화살촉

```
\put(25,25){\vector(1,2){25}}  
\put(25,25){\vector(-1,-1){20}}  
\put(25,25){\vector(0,1){35}}
```

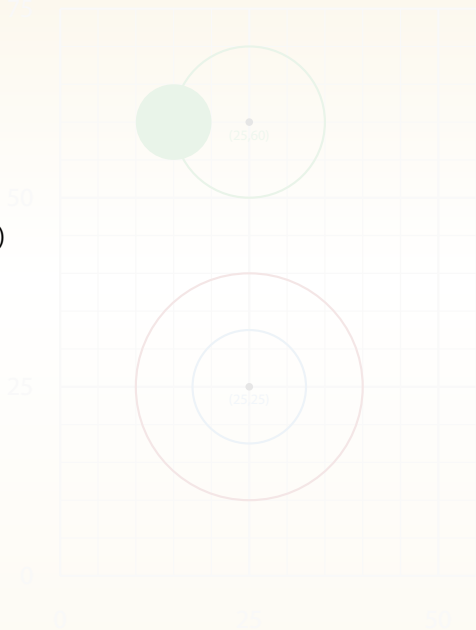


# \circle 원

```
\circle{지름}  
\circle*{지름}
```

- \*-표가 붙으면 속을 채운 원 (disk)
- 반지름이 아니라 **지름!**

```
\put(25,25){\circle{30}}  
\put(25,25){\circle{15}}  
\put(15,60){\circle*{10}}  
\put(25,60){\circle{20}}  
  
\put(25,60){\circle*{1}}
```

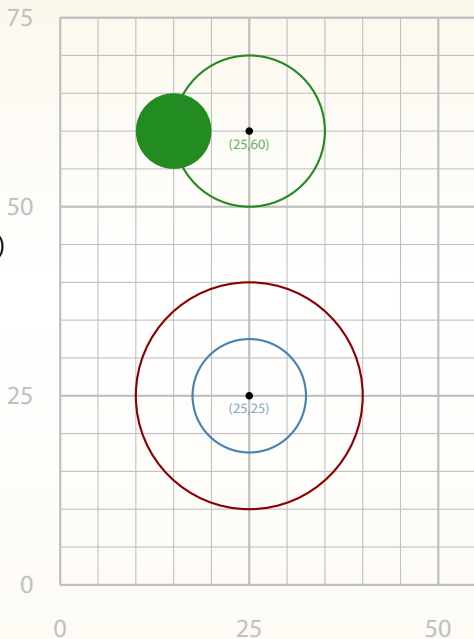


# \circle 원

```
\circle{지름}  
\circle*{지름}
```

- \*-표가 붙으면 속을 채운 원 (disk)
- 반지름이 아니라 **지름!**

```
\put(25,25){\circle{30}}  
\put(25,25){\circle{15}}  
\put(15,60){\circle*{10}}  
\put(25,60){\circle{20}}  
  
\put(25,60){\circle*{1}}
```



## \oval 오벌

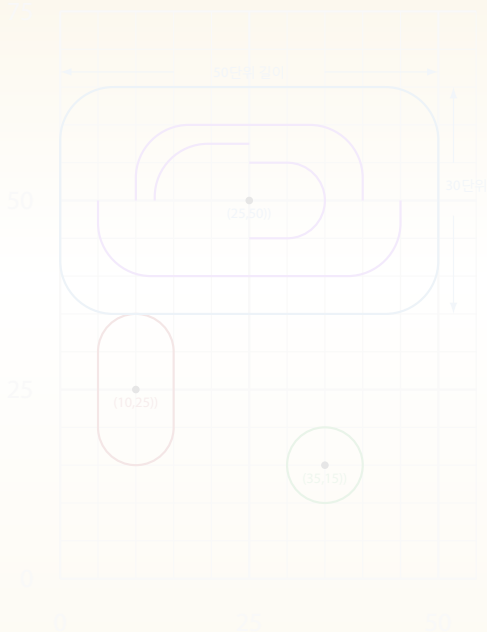
`\oval(x,y) [t/b/l/r]`

- 모서리를 둥글린 사각형
- $x, y$ 는 사각형의 가로/세로 길이
- $t/b/l/r$ 은 상하좌우의 일부분

```
\put(10,25){\oval(10,20)}  
\put(25,50){\oval(50,30)}  
\put(35,15){\oval(10,10)}
```

일부만 그리기

```
\put(25,50){\oval(30,20)[t]}  
\put(25,50){\oval(40,20)[b]}  
\put(25,50){\oval(20,10)[r]}  
\put(25,50){\oval(25,15)[t1]}
```



## \oval 오벌

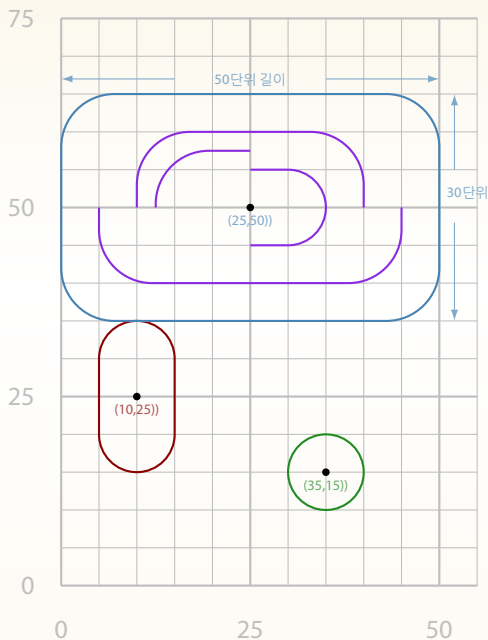
`\oval(x,y) [t/b/l/r]`

- 모서리를 둥글린 사각형
- $x, y$ 는 사각형의 가로/세로 길이
- $t/b/l/r$ 은 상하좌우의 일부분

```
\put(10,25){\oval(10,20)}  
\put(25,50){\oval(50,30)}  
\put(35,15){\oval(10,10)}
```

일부만 그리기

```
\put(25,50){\oval(30,20)[t]}  
\put(25,50){\oval(40,20)[b]}  
\put(25,50){\oval(20,10)[r]}  
\put(25,50){\oval(25,15)[t1]}
```



## `\qBezier` 쿼드러틱 베지어 곡선

```
\qBezier(x1,y1)(x2,y2)(x3,y3)
```

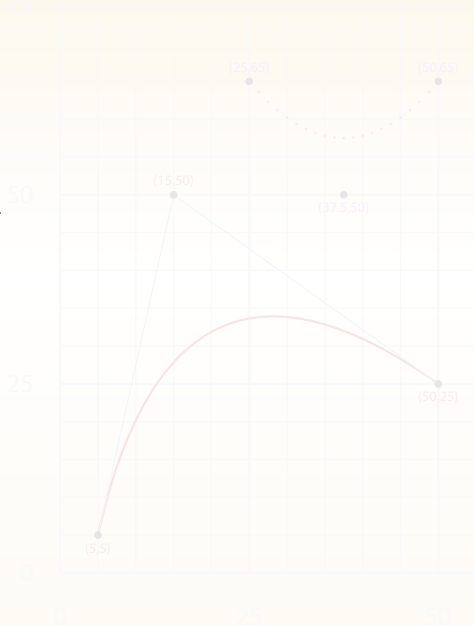
```
\qBezier[점의 개수]
```

```
(x1,y1)(x2,y2)(x3,y3)
```

- 시점  $(x_1, y_1)$  과 종점  $(x_3, y_3)$  를 부드럽게 잇는 곡선
- $(x_2, y_2)$  는 컨트롤 포인트
- [점의 개수]  $\Rightarrow$  이 숫자 만큼 점을 찍어 곡선 표현

```
\qBezier(5,5)(15,50)(50,25)
```

```
\qBezier[20](5,5)(15,50)(50,25)
```





# \qBezier 쿼드러틱 베지어 곡선

75

```
\qBezier(x1,y1)(x2,y2)(x3,y3)
```

```
\qBezier[점의 개수]
```

```
(x1,y1)(x2,y2)(x3,y3)
```

- 시점  $(x_1, y_1)$  과 종점  $(x_3, y_3)$  를 부드럽게 잇는 곡선
- $(x_2, y_2)$  는 컨트롤 포인트
- [점의 개수]  $\Rightarrow$  이 숫자 만큼 점을 찍어 곡선 표현

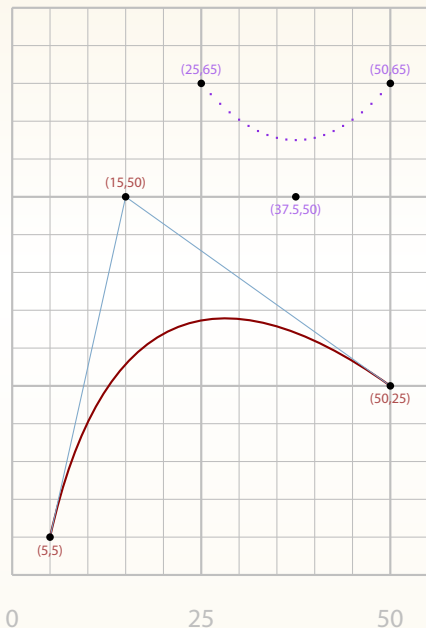
```
\qBezier(5,5)(15,50)(50,25)
```

```
\qBezier[20](5,5)(15,50)(50,25)
```

50

25

0



0

25

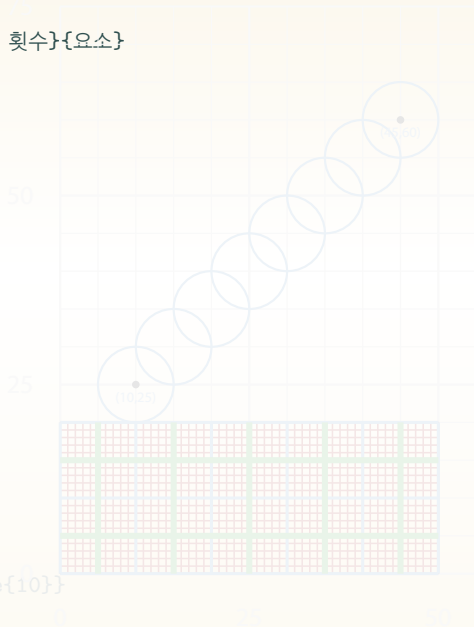
50

## `\multiput` 반복되는 패턴

`\multiput(x,y)(x-증분,y-증분){반복 횟수}{요소}`

- 일정한 위치 변화와 모양 반복
- 시점  $(x,y)$  를 기준으로  $(\Delta x, \Delta y)$  만큼씩 이동

```
... \linethickness{0.2mm}
\multiput(0,0)(1,0){51}{%
\line(0,1){20}}
... \linethickness{0.4mm}
\multiput(0,0)(5,0){11}{%
\line(0,1){20}}
... \linethickness{.8mm}
\multiput(5,0)(10,0){5}{%
\line(0,1){20}}
\multiput(10,25)(5,5){8}{\circle{10}}
```



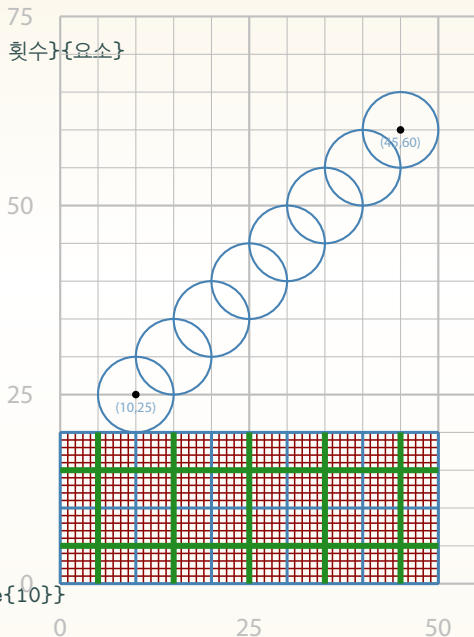
## `\multiput` 반복되는 패턴

`\multiput(x,y)(x-증분,y-증분){반복 횟수}{요소}`

- 일정한 위치 변화와 모양 반복
- 시점  $(x,y)$  를 기준으로  $(\Delta x, \Delta y)$  만큼씩 이동

```
... \linethickness{0.2mm}
\multiput(0,0)(1,0){51}{%
\line(0,1){20}}
... \linethickness{0.4mm}
\multiput(0,0)(5,0){11}{%
\line(0,1){20}}
... \linethickness{.8mm}
\multiput(5,0)(10,0){5}{%
\line(0,1){20}}
```

```
\multiput(10,25)(5,5){8}{\circle*{10}}
```

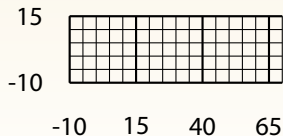


## `\graphpaper` 그리드

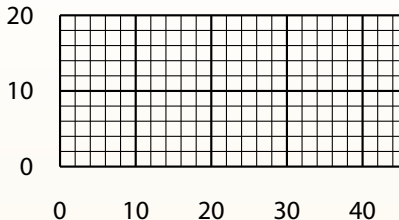
```
\usepackage{graphpap}  
...  
\graphpaper(x1,y1)(x2,y2)  
\graphpaper[모눈 간격](x1,y1)(x2,y2)
```

- Built-in은 아니지만...
- $(x_1, y_1)$ : Lower Left
- $(x_2, y_2)$ : Upper Right
- 기본 10단위마다 모눈 생성
- [모눈 간격] 숫자마다 모눈 그림

```
\setlength\unitlength{1pt}  
\begin{picture}(80,25)(-20,-20)  
\graphpaper[5](-10,-10)(80,25)  
\end{picture}
```



```
\setlength\unitlength{1mm}  
\begin{picture}(45,20)  
\graphpaper[2](0,0)(45,20)  
\end{picture}
```



## 슬픈 사연

- `\picture`의 애처로울 정도로 불완전한 기능
  - 한정된 `\line` 방향(기울기)  
 $-6 \leq (x, y) \leq 6$ , where  $(x, y) \in \mathbb{Z}$
  - `\vector`의 사정은 더욱 악화  
 $-4 \leq (x, y) \leq 4$ , where  $(x, y) \in \mathbb{Z}$
  - `\circle`의 최대 지름 40pt ( $\approx 14\text{mm}$ )
  - `\circle*`의 최대 지름 15pt ( $\approx 5\text{mm}$ )
- 이걸로 대체 뭘 할 수 있단 말이지?

## 슬픈 사연

- `\picture`의 애처로울 정도로 불완전한 기능
  - 한정된 `\line` 방향(기울기)  
 $-6 \leq (x, y) \leq 6, \text{ where } (x, y) \in \mathbb{Z}$
  - `\vector`의 사정은 더욱 악화  
 $-4 \leq (x, y) \leq 4, \text{ where } (x, y) \in \mathbb{Z}$
  - `\circle`의 최대 지름 40pt ( $\approx 14\text{mm}$ )
  - `\circle*`의 최대 지름 15pt ( $\approx 5\text{mm}$ )
- 이걸로 대체 뭘 할 수 있단 말ियो?

## 왜 이런 일이 생겼을까?

- $\text{\TeX}$ 의 폰트 구현 방식: METAFONT
- 당시 직선과 원은 폰트로 삽입
- 기울기가 다른 선분과 원(호)를 적당히 분할하여 폰트에 글리프로 심어  
놓음
- 하나의 .mf에 담을 수 있는 글리프는 최대 몇 개? 너무나 익숙한 128개  
⇒ 이걸로 모든 기울기를 구현하라급쇼?
- 원은 `line10.mf`, 선은 `lcircle10.mf`

## 왜 이런 일이 생겼을까?

- T<sub>E</sub>X의 폰트 구현 방식: METAFONT
- 당시 직선과 원은 폰트로 삽입
- 기울기가 다른 선분과 원(호)를 적당히 분할하여 폰트에 글리프로 심어  
놓음
- 하나의 .mf에 담을 수 있는 글리프는 최대 몇 개? 너무나 익숙한 128개  
⇒ 이걸로 모든 기울기를 구현하라급쇼?
- 원은 `line10.mf`, 선은 `lcircle10.mf`



## 왜 이런 일이 생겼을까?

- T<sub>E</sub>X의 폰트 구현 방식: METAFONT
- 당시 직선과 원은 폰트로 삽입
- 기울기가 다른 선분과 원(호)를 적당히 분할하여 폰트에 글리프로 심어  
놓음
- 하나의 .mf에 담을 수 있는 글리프는 최대 몇 개? 너무나 익숙한 128개  
⇒ 이걸로 모든 기울기를 구현하라급쇼?
- 원은 `line10.mf`, 선은 `lcircle10.mf`

## 왜 이런 일이 생겼을까?

- T<sub>E</sub>X의 폰트 구현 방식: METAFONT
- 당시 직선과 원은 폰트로 삽입
- 기울기가 다른 선분과 원(호)를 적당히 분할하여 **폰트**에 글리프로 심어  
놓음
- 하나의 .mf에 담을 수 있는 글리프는 최대 몇 개? 너무나 익숙한 128개  
⇒ 이걸로 모든 기울기를 구현하라급쇼?
- 원은 `line10.mf`, 선은 `lcircle10.mf`

## 왜 이런 일이 생겼을까?

- T<sub>E</sub>X의 폰트 구현 방식: METAFONT
- 당시 직선과 원은 폰트로 삽입
- 기울기가 다른 선분과 원(호)를 적당히 분할하여 **폰트**에 글리프로 심어  
놓음
- 하나의 .mf에 담을 수 있는 글리프는 최대 몇 개? 너무나 익숙한 128개  
⇒ 이걸로 모든 기울기를 구현하라급쇼?
- 원은 **line10.mf**, 선은 **lcircle10.mf**

# line10.mf 폰트 차트

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x	/	/	/	/	/	/			"0x
'01x	/	▲	/	▶	/	▼	▲	▼	
'02x	/	/	▼	/	/	▲		▲	"1x
'03x	/	▲	/	▲	/	▶	▲		
'04x	/	/	/	/		/			"2x
'05x	/	▲	▼	▲	/	▶		▶	
'06x		▼		▼			▲	▲	"3x
'07x		▲	▼			▲	▼	▼	
'10x	\	\	\	\	\	\			"4x
'11x	\	▼	\	▶	\	▶	▼	▶	
'12x	\	\	▲	\	\	▲		▼	"5x
'13x	\	▼	\		\	▶	▲		
'14x	\	\	\	\		\			"6x
'15x	\	▼	▲	▼	\			▶	
'16x		▲		▲				▲	"7x
'17x		▼	▶			▼	▲		
	"8	"9	"A	"B	"C	"D	"E	"F	

# lcircle10.mf 폰트 차트

	'0	'1	'2	'3	'4	'5	'6	'7	
'00x									"0x
'01x									
'02x									"1x
'03x									
'04x									"2x
'05x									
'14x									"6x
'15x									
'16x									"7x
'17x									
	"8	"9	"A	"B	"C	"D	"E	"F	

# 1사분면에서 \line으로 구현할 수 있는 모든 방향 벡터 쌍

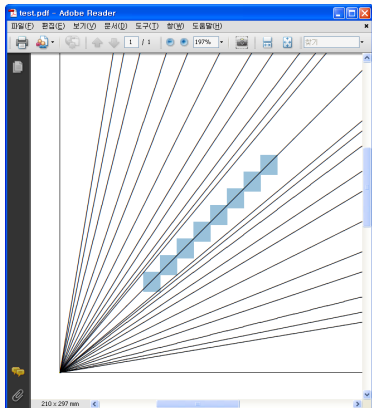
$(x, y)$	0	1	2	3	4	5	6	개수
0	<del>(0,0)</del>	(0,1)	<del>(0,2)</del>	<del>(0,3)</del>	<del>(0,4)</del>	<del>(0,5)</del>	<del>(0,6)</del>	1
1	(1,0)	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	7
2	<del>(2,0)</del>	(2,1)	<del>(2,2)</del>	(2,3)	<del>(2,4)</del>	(2,5)	<del>(2,6)</del>	3
3	<del>(3,0)</del>	(3,1)	(3,2)	<del>(3,3)</del>	(3,4)	(3,5)	<del>(3,6)</del>	4
4	<del>(4,0)</del>	(4,1)	<del>(4,2)</del>	(4,3)	<del>(4,4)</del>	(4,5)	<del>(4,6)</del>	3
5	<del>(5,0)</del>	(5,1)	(5,2)	(5,3)	(5,4)	<del>(5,5)</del>	(5,6)	5
6	<del>(6,0)</del>	(6,1)	<del>(6,2)</del>	<del>(6,3)</del>	<del>(6,4)</del>	(6,5)	<del>(6,6)</del>	2
개수	1	7	3	4	3	5	2	25

- $-6 \leq x, y \leq 6$ , 단  $x, y$ 는 정수,  $x, y$ 는 서로소 (relatively prime)

# 그릴 수 있는 모든 직선의 기울기

- 1사분면 25개 ( $x$ -축,  $y$ -축 포함)

```
\setlength{\unitlength}{5cm}
\begin{picture}(1, 1)
  \put(0, 0){\line(0, 1){1}}
  \put(0, 0){\line(1, 0){1}}
  \put(0, 0){\line(1, 1){1}}
  \put(0, 0){\line(1, 2){.5}}
  \put(0, 0){\line(1, 3){.33333}}
  ...
  \put(0, 0){\line(5, 6){.83333}}
  \put(0, 0){\line(6, 1){1}}
  \put(0, 0){\line(6, 5){1}}
\end{picture}
```



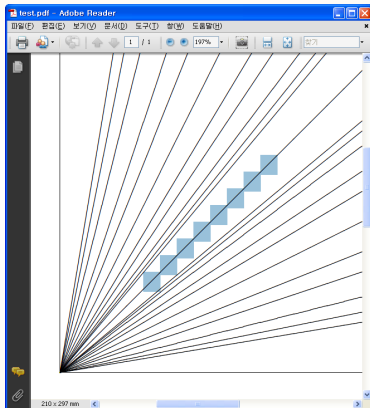
[잠깐 퀴즈] 그렇다면 2차 평면 전체에

구현할 수 있는 기울기의 개수는 모두 몇 개일까?

# 그릴 수 있는 모든 직선의 기울기

- 1사분면 25개 ( $x$ -축,  $y$ -축 포함)

```
\setlength{\unitlength}{5cm}
\begin{picture}(1, 1)
  \put(0, 0){\line(0, 1){1}}
  \put(0, 0){\line(1, 0){1}}
  \put(0, 0){\line(1, 1){1}}
  \put(0, 0){\line(1, 2){.5}}
  \put(0, 0){\line(1, 3){.33333}}
  ...
  \put(0, 0){\line(5, 6){.83333}}
  \put(0, 0){\line(6, 1){1}}
  \put(0, 0){\line(6, 5){1}}
\end{picture}
```



[잠깐 퀴즈] 그렇다면 2차 평면 전체에

구현할 수 있는 기울기의 개수는 모두 몇 개일까?



# 지금 설명할 것은

## picture 환경

LaTeX에서 기본으로 제공하는 '간단한' 그리기 환경

## epic, eepic, curves 패키지

picture의 한계를 뛰어 넘으려는 노력

## pict2e 패키지

11년만에 완성된 '간단한' LaTeX드로잉 패키지

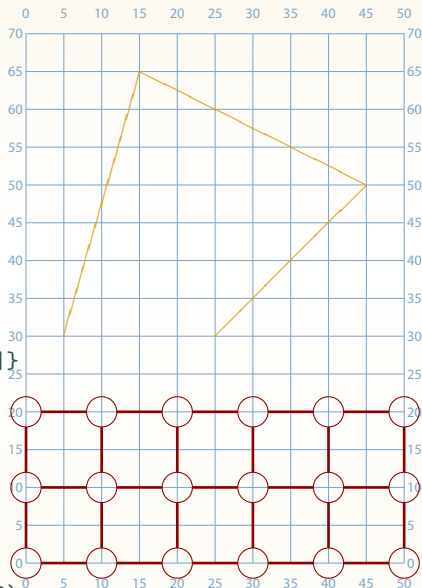
## curve2e 패키지

LaTeX, 곡선에서 놀다

# Hello, epic?

- 좌표 평면을 손쉽게 그리는  
`\grid`
- 더 게을러진(?) 반복 작업  
`\matrixput`
- 좌표점들을 연결해 선을 그리는  
`\drawline` 등

```
\put(0,0){\tiny \grid(50,70)(5,5)[0,0]}
\matrixput(0,0)(10,0){6}(0,10){3}{%
\circle{4}}
  \linethickness{1t}
\matrixput(2,0)(10,0){5}(0,10){3}{%
  \line(1,0){6}}
\matrixput(0,2)(10,0){6}(0,10){2}{%
  \line(0,1){6}}
\drawline[1](5,30)(15,65)(45,50)(25,30)
```

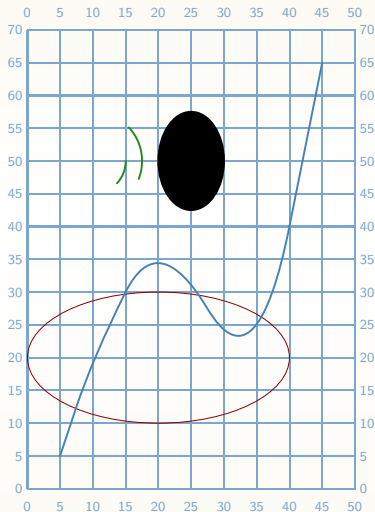


# Hello, eepic?

- 구현할 수 있는 기울기에 제한이 없어짐
- 구현할 수 있는 원과 disk의 지름 크기에 제한이 없어짐
- `\drawline` 대신 `\path`
- 스플라인 곡선: `\spline`
- 타원: `\ellipse`, `\ellipse*`
- 호: `\arc`

패키지 충돌로 이 예제는 다른 곳에서

그려 불러옴



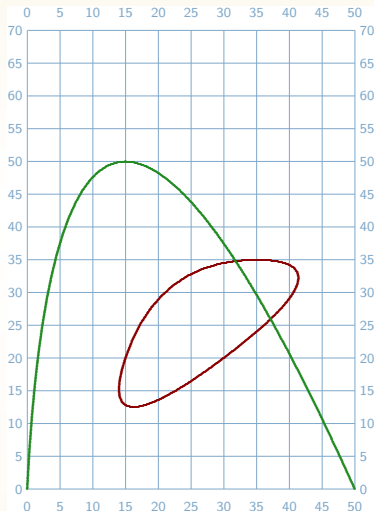
# Hello, curves?

- 회전, 확대/축소 등 변환 가능
- `\curve`
- `\closedcurve`
- `\tagcurve`

패키지 충돌로 이 예제는 다른 곳에서  
그려 불러옴

```
\put(0,0){%  
  \closedcurve(15,20,35,35,30,20)}
```

```
\put(0,0){\curve(0,0,15,50,50,0)}
```



# 지금 설명할 것은

## picture 환경

L<sup>A</sup>T<sub>E</sub>X에서 기본으로 제공하는 '간단한' 그리기 환경

## epic, eepic, curves 패키지

picture의 한계를 뛰어 넘으려는 노력


## pict2e 패키지

11년만에 완성된 '간단한' L<sup>A</sup>T<sub>E</sub>X드로잉 패키지

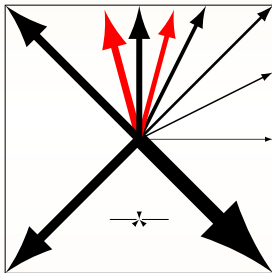
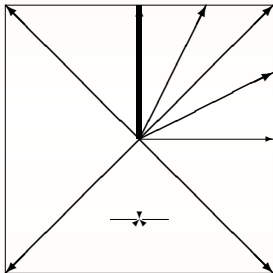
## curve2e 패키지

L<sup>A</sup>T<sub>E</sub>X, 곡선에서 놀다

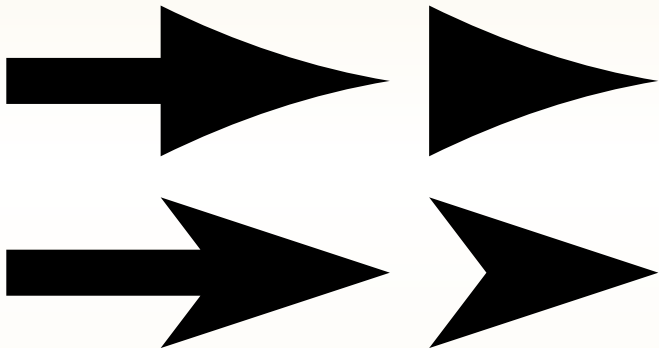
## Wow, pict2e!

- `picture` 환경의 한계를 극복하려는 그간의 노력을 집대성한 패키지
  - *LaTeX: A Document Preparation System*, 2nd Ed.에서 언급 (1993)
  - 11년만에 공식 발표 (2003)
- 많은 그래픽 명령어를 새로 정의
  - 기울기는 서로소가 아니어도 상관없음  $\rightarrow -1000 \leq (x, y) \leq 1000$
  - 선 굵기 제약 없음
  - 화살표 모양 선택 가능 (`ltxarrows/pstarrows`)
  - `\maxovalrad`: 오벌의 모서리 둥글리기 값 제어
  - `\cbezier`: 큐빅 베지어 커브 ( $\leftarrow$  `\bezier`, `\qbezier`)
  - 간단해진 `\arc` 명령  
`\arc[0,75]{10} \arc*[180,135]{10} \Rightarrow` 

# pict2e 구경하기 I

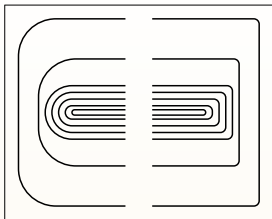
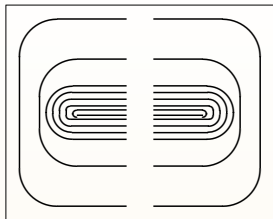
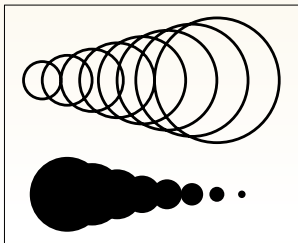
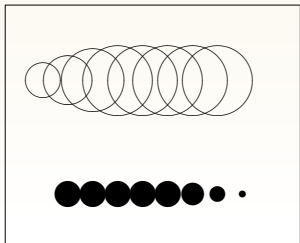


pict2e 구경하기 II



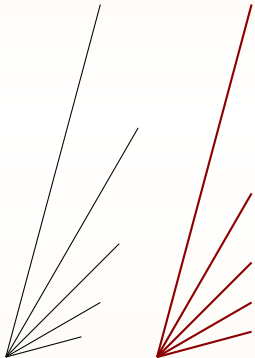


# pict2e 구경하기 III



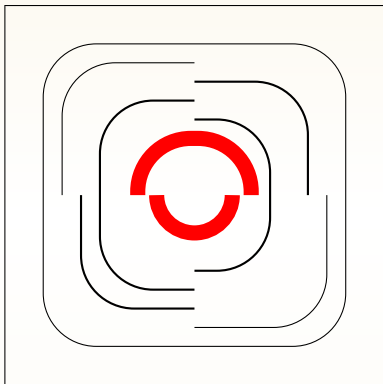
## pict2e 구경하기 IV

- 기울기의 재정의



```
\setlength{\unitlength}{.5mm}
\begin{picture}(80,50)(30,30)
\begin{picture}(80,50)(30,30)
\put(40,0){\line(966,259){20}}
\put(40,0){\line(866,500){25}}
\put(40,0){\line(707,707){30}}
\put(40,0){\line(500,866){35}}
\put(40,0){\line(259,966){25}}
\color{DarkRed}
\put(80,0){\line(966,259){25}}
\put(80,0){\line(866,500){25}}
\put(80,0){\line(707,707){25}}
\put(80,0){\line(500,866){25}}
\put(80,0){\line(259,966){25}}
\end{picture}
\end{picture}
```

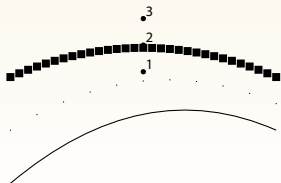
## pict2e 구경하기 V



- oval의 모서리 조정

```
\begin{picture}(100,100)%  
\thinlines  
\put(0,0){\framebox(100,100){}}  
\put(50,50){\oval(80,80)}  
\put(50,50){\oval(70,70)[t1]}  
\put(50,50){\oval(70,70)[br]}  
\thicklines  
\put(50,50){\oval(60,60)[tr]}  
\put(50,50){\oval(60,60)[bl]}  
\put(50,50){\oval(50,50)[l]}  
\put(50,50){\oval(40,40)[r]}  
\linethickness{2mm} \color{red}  
\put(50,50){\oval(30,30)[t]}  
\put(50,50){\oval(20,20)[b]}  
\end{picture}
```

## pict2e 구경하기 VI



- 쿼드러틱 베지어 커브

```
\put(50,42){\circle*{2}}
```

```
\put(51,43){\tiny1}
```

...

```
\qBezier(0,0)(50,42)(100,20)
```

```
\qBezier[10](0,20)(50,52)(100,30)
```

...

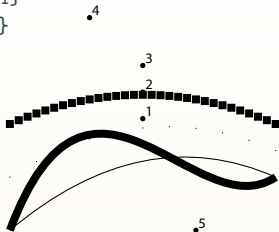
```
% cubic bezier
```

```
\put(30,80){\circle*{2}} \put(31,81){\tiny4}
```

```
\put(70,0){\circle*{2}} \put(71,1){\tiny5}
```

```
\cBezier(0,0)(30,80)(70,0)(100,20)
```

- 큐빅 베지어 커브



# 지금 설명할 것은

## picture 환경

$\text{\LaTeX}$ 에서 기본으로 제공하는 '간단한' 그리기 환경

## epic, eepic, curves 패키지

picture의 한계를 뛰어 넘으려는 노력

## pict2e 패키지

11년만에 완성된 '간단한'  $\text{\LaTeX}$ 드로잉 패키지

## curve2e 패키지

$\text{\LaTeX}$ , 곡선에서 놀다

## Awesome, `curve2e`!

- 이제  $\LaTeX$ 에서 뚝뚝하게 그림을 그린다고 말할 수 있다!

- `pict2e` 패키지를 더 개선

- 많은 그래픽 명령어를 새로 정의

- `\LINE(x1,y1)(x2,y2)`: 두 점을 그냥 이어버려!

- `\VECTOR(x1,y1)(x2,y2)`

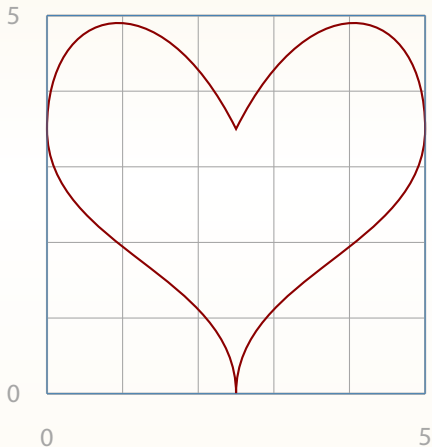
- 간단해진 `\Arc` 명령 `\Arc(0,0)(1,0){45}`  $\Rightarrow$



- `\Curve`: 방향과 극점

- `\cbezier`: 큐빅 베지어 커브 ( $\Leftarrow$  `\bezier`, `\qbezier`)

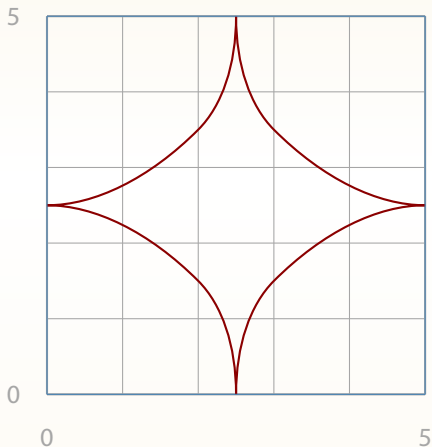
## curve2e 구경하기 I



- 뾰족한 점(cusp)이 있는 곡선

```
\setlength\unitlength{1cm}
\begin{picture}(5,5)(0.,0.)
  \Curve(2.5,0)<0,1>(5,3.5)<0,1>%
  (2.5,3.5)<-.5,-1>[-.5,1]%
  (0,3.5)<0,-1>(2.5,0)<0,-1>
  \color{SteelBlue}
  \thinlines
  \put(0,0){\line(0,5){5}}
  \put(0,0){\line(5,0){5}}
  \put(5,0){\Line(0,5)}
  \put(0,5){\line(1,0){5}}
\end{picture}
```

## curve2e 구경하기 II



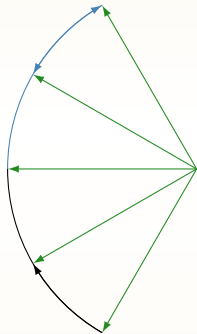
- 뾰족한 점 (cusp) 이 있는 곡선

```
\setlength\unitlength{1cm}
\begin{picture}(5,5)(0.,0.)
\Curve(2.5,0.0)<0,1>(3.0,1.5)<1,1>%
(5.0,2.5)<1,0>[-1,0](3.0,3.5)<-1,1>%
(2.5,5.0)<0,1>[0,-1](2.,3.5)<-1,-1>%
(0.0,2.5)<-1,0>[1,0](2.,1.5)<1,-1>%
(2.5,0.0)<0,-1>
\color{SteelBlue}
\thinlines
\put(0,0){\line(0,5){5}}
\put(0,0){\line(5,0){5}}
\put(5,0){\Line(0,5)}
\put(0,5){\line(5,0){5}}
\end{picture}
```



## curve2e 구경하기 III

- 호(arc)와 멀티벡터



```
\setlength\unitlength{1cm}
\begin{picture}(2.5,5)(-2.5,-2.5)
  \DivideE 360pt by 12pt to\Twelfth
  \DirFromAngle\Twelfth to\Dir
  \CopyVect -2.5,0 to\Vzero
  \MultVect\Vzero by\Dir to\Vone \MultVect\Vone by
  \Arc(0.0,0.0)(\Vzero){30}
  \Arc(0.0,0.0)(\Vone){30}
  \VectorArc(0.0,0.0)(\Vtwo){-30}
  \color{SteelBlue}
  \MultVect\Vzero by*\Dir to\Vmone \MultVect\Vmone
  \Arc(0.0,0.0)(\Vzero){-30} \Arc(0.0,0.0)(\Vmone
  \VectorARC(0.0,0.0)(\Vtwo){30}
  \color{ForestGreen}\thinlines
  \VECTOR(0,0)(\Vzero)
  \VECTOR(0,0)(\Vone)\VECTOR(0,0)(\Vmone)
  \VECTOR(0,0)(\Vtwo)\VECTOR(0,0)(\Vtwo)
\end{picture}
```

숙제를 잊진 않았겠죠?

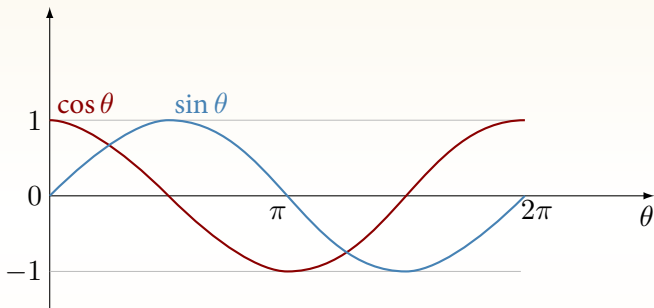


Figure:  $\sin \theta$ 와  $\cos \theta$ 의 그래프 ( $0 \leq \theta \leq 2\pi$ )

- 풀이를 KTUG 게시판에 올려주세요.

## 단위 길이를 1cm로 했을 때

```
\setlength\unitlength{1cm}
\begin{picture}(8,4)(0,-1.5)
% \put(0,-1.5){\Vector(0,1){4}}
\VECTOR(0,-1.5)(0,2.5)
\put(0,0){\Vector(8,0)}
\put(7.8,-.4){$\theta$}
\put(-0.1,1){\makebox(0,0)[r]{$1$}}% 1
\put(-0.1,0){\makebox(0,0)[r]{$0$}}% 1
\put(-0.1,-1){\makebox(0,0)[r]{$-1$}}% -1
\put(2.9,-0.3){$\pi$}% pi
\put(6.23,-0.3){$2\pi$}% 2 pi
\color{Gray!50}
\put(0,1){\line(1,0){6.23}}      \put(0,-1){\Line(6.23,0)}
\thicklines % 핵심은 여기
\color{DarkRed}\put(0.1,1.1){$\cos\theta$}
\Curve(0,1)<1,0>(1.570796,0)<1,-1>%
(3.1415924,-1)<1,0>(6.283185,1)<1,0>%
\color{SteelBlue}\put(1.65,1.1){$\sin\theta$}
\Curve(0,0)<1,1>(1.570796,1)<1,0>%
(4.712389,-1)<1,0>(6.283185,0)<1,1>%
\end{picture}
```

## 마치며

- 설명 드린 모든 패키지를 불러올 필요는 없어요.
- 그래픽 틀은 위지웍 방식이든 랭귀지 코딩이든 익숙한 것을 쓰세요.
- KTUG 게시판에 아는 질문이 나오면 적극적으로 답변을 달아주세요.
- Happy  $\TeX$ ing!

## 마치며

- 설명 드린 모든 패키지를 불러올 필요는 없어요.
- 그래픽 툴은 위지윅 방식이든 랭귀지 코딩이든 익숙한 것을 쓰세요.
- KTUG 게시판에 아는 질문이 나오면 적극적으로 답변을 달아주세요.
- Happy T<sub>E</sub>Xing!

## 마치며

- 설명 드린 모든 패키지를 불러올 필요는 없어요.
- 그래픽 툴은 위지윅 방식이든 랭귀지 코딩이든 익숙한 것을 쓰세요.
- KTUG 게시판에 아는 질문이 나오면 적극적으로 답변을 달아주세요.
- Happy T<sub>E</sub>Xing!

## 마치며

- 설명 드린 모든 패키지를 불러올 필요는 없어요.
- 그래픽 툴은 위지윅 방식이든 랭귀지 코딩이든 익숙한 것을 쓰세요.
- KTUG 게시판에 아는 질문이 나오면 적극적으로 답변을 달아주세요.
- Happy  $\TeX$ ing!

# L<sup>A</sup>T<sub>E</sub>X Graphics

## L<sup>A</sup>T<sub>E</sub>X Graphics

• 2010년 11월 6일 (토) 09:00

- 공주대학교 인문사회관 컴퓨터실  
107호 (Session 1)  
108호 (Session 2, 3)



### 진행순서

09:00-09:30	등록
09:30-12:00	Session 1. L <sup>A</sup> T <sub>E</sub> X 기초 ..... 107호 사회: 김영록 한국외국어대 • L <sup>A</sup> T <sub>E</sub> X 소개 및 기본 사용방법 조진환 수원대
10:00-12:00	Session 2. 그림의 거의 모든 것 ..... 108호 사회: 김도현 동국대 • 그림을 준비하고 문서에 포함하기 이호재 AST • 그림의 형식과 적절한 배치
12:00-13:00	점심
13:00-17:00	Session 3. 페이지 위에 그림 그리기 ... 108호 사회: 이기황 연세대
13:00-14:00	• Beamer에서 TikZ 활용 조영철 고려대
14:00-14:40	• L <sup>A</sup> T <sub>E</sub> X의 기본 picture 환경 이주호 국회예산정책처
15:00-16:00	• PSTricks 맛보기/papicture 환경 김강수 KRUG
16:00-17:00	• TikZ로 페이지 꾸미기/tikpicture 환경 김강수 KRUG
17:00-17:30	질의와 답변

