

# TikZ 기초

박승원

<http://swpark.me>

서울대학교 물리천문학부

2017년 11월 18일

# 목적

TikZ (틱즈?)

## 강의 목적

- ▶ TikZ가 처음인  $\text{\LaTeX}$  사용자를 대상으로,
  - ▶ 최대한 간추려서,
  - ▶ 뒤에 있을 강의들을 따라갈 수 있도록.
  - ▶ + 원하는 그림을 TikZ로 그릴 수 있게.

## 본 자료의 목적

- ▶ 워크숍 종료 후 참고자료로 활용
- ▶ 워크숍 비참여자를 위한 지침서

# 목차

- ▶ 목적 / 목차
- ▶ TikZ 개관
- ▶ Main: TikZ 주요 문법
  - ▶ 사용 빈도 순으로 소개.
- ▶ TikZ 사용 팁 / 관련 논의
- ▶ 읽을거리 / 참고문헌

# TikZ 개관

- ▶ “**TikZ ist kein Zeichenprogramm**”
  - ▶ “TikZ is not a *drawing* program”
  - ▶ 마우스/태블릿으로 그리는 것이 아니다!
- ▶ T<sub>E</sub>X에서 그림 그리기, 판면 꾸미기 등에 사용.
- ▶ PSTricks 이후로 TikZ가 대세.

# 그림 도구로서의 TikZ

자주 쓰이는 그림 도구들:

- ▶ PowerPoint / 아래아한글
- ▶ Adobe Illustrator
- ▶ **GeoGebra**
- ▶ **TikZ**
- ▶ **Inkscape** / **GIMP** / Windows 그림판
- ▶ **<https://draw.io>**
- ▶ 손으로 그리기

(볼드: 자유 소프트웨어.)

# TikZ 준비하기

template.tex ← (examples 폴더에 있습니다.)

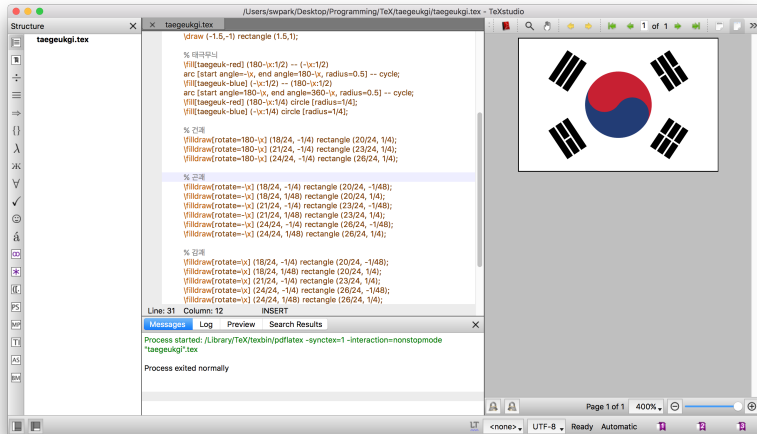
- ▶ 그림을 그리기 위한 기본 템플릿.

```
1 \documentclass{standalone}
2 \usepackage{tikz}
3 \begin{document}
4     \begin{tikzpicture}
5         \draw (0,0) -- (1,1); % contents goes here
6     \end{tikzpicture}
7 \end{document}
```

- ▶ standalone 클래스의 사용
  - ▶ tikzpicture 외에는 아무것도 없어야 함.
- ▶ 옆에 미리보기 창을 띄워 놓고 수시로 컴파일해보며 작업.

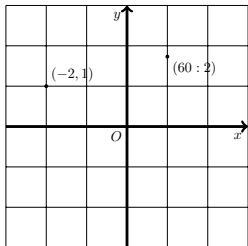
# TikZ 준비하기

TeXstudio의 경우, 이렇게...



# TikZ 좌표계

- ▶ 오늘은 2차원 데카르트 좌표계, 극좌표계만 사용한다.
- ▶ 3차원, 택시거리계, 무게중심 좌표계, ...
  - ▶ pgfmanual(TikZ 매뉴얼) 13장 참조
  - ▶ 여기에서부터 pgfmanual이라는 말은 생략.





# 들어가기에 앞서

intro.tex

아래의 코드는 어떤 그림을 그릴까요?

```
1 \draw (0,0) -- (1,1);  
2 \draw (0,0) rectangle (2,1);  
3 \draw (0,0) circle [radius=0.5];  
4 \node at (0,0) {Some text};  
5 \draw (0,0) -- (120:2);
```

- ▶ 간결하고 알아보기 쉬운 TikZ의 문법.
- ▶ 보통의  $\text{\LaTeX}$ 과 다르게, 각 명령은 세미콜론(;)으로 종결.

# 경로 그리기

path.tex

```
1 \draw (0,0) -- (1,1);  
2 \draw (0,0) -- (1,-1) -- (2,-2);  
3 \draw (0,0) -- (2,0) -- (2,3) -- (0,3) -- cycle;  
4 \draw (3,0) arc (0:120:3) -- cycle;
```

- ▶ cycle: 첫번째/마지막 점을 이어 폐곡선을 만듦
- ▶ -- 외에도,
  - ▶ -,|, arc, parabola, sin, ...
  - ▶ visualtikz 2장 참조.

# 도형 그리기

shapes.tex

```
1 \draw (0,0) circle [radius=0.5];  
2 \draw (0,0) rectangle (2,3);  
3 \draw (0,0) grid (5,5);
```

- ▶ 엄밀히는 도형도 'path'.
- ▶ 따라서 도형 내부는 투명.

# 경로 꾸미기

path-style.tex

```
1 \draw [옵션1, 옵션2] 경로;  
2 \draw[blue, <->, dashed] (0,0) -- (4,1) -- (3,3);  
3 \draw (1,2) -- (2,2) [->] -- (3,1) -- (4,2); % Nope.
```

- ▶ 색: red, blue, white, ...
- ▶ 두께: line width = 2pt, thick, ... (15.3.1절)
- ▶ 형태: dashed, dotted, ...
- ▶ 화살표 축: ->, <->, |<->| 등 자유롭게
  - ▶ 더 다양한 축 모양은 visualtikz 4.10절 참조
  - ▶ (번거로우니) 웬만하면 그냥 쓰시다.

# 이런 것도 있습니다

coil.tex

- ▶ 경로의 형태는 정말 다양한 것이 가능.
  - ▶ ex) 코일: (와, PPT에는 없는 도구다!)
- ▶ `\usetikzlibrary{...}`는 preamble<sup>1</sup>에 삽입.

```
1 % \usetikzlibrary{decorations.pathmorphing}  
2 \draw[decoration={segment length=2mm, amplitude=2mm,  
  ↪ coil}, decorate] (0,0) -- (2,2);
```

---

<sup>1</sup>`\begin{document}` 이전

# 도형 꾸미기

shapes-style.tex

- ▶ 테두리 조정: 경로와 동일
- ▶ 색 채우기( $\backslash$ draw는 투명):  $\backslash$ fill,  $\backslash$ filldraw
- ▶ 무늬: (60장 참조)
  - ▶  $\backslash$ usetikzlibrary{patterns}
  - ▶ 'north east lines', 'dots', 'bricks', ...

```
1 % \usetikzlibrary{patterns}
2 \fill[red] (0,0) circle [radius=1];
3 \filldraw[pattern=bricks, draw=blue, ultra thick] (0,0)
   ↪ rectangle (2,3);
4 \filldraw[fill=green, opacity=0.8, draw=blue, ultra
   ↪ thick] (1,1) rectangle (3,4);
```

# 텍스트 삽입

text.tex

새로운 명령어: `\node`

```
1 \node at (0,2) {Some text};  
2 \draw (2,-1) -- (2,2) node[right] {A};  
3 \draw[|<->|] (0,0) -- node[midway, above] {$ 1 $} (3,1);
```

텍스트 대신 그림도 삽입할 수 있다.

# 상대적 위치 사용하기

relative.tex

아래 셋은 모두 같은 결과를 낸다.

```
1 \draw (10,10) -- (10,11) -- (11,11);  
2 \draw (10,10) -- +(0,1) -- +(1,1);  
3 \draw (10,10) -- ++(0,1) -- ++(1,0);
```

- ▶ ‘+’ 1개: 임시적. 뒤의 상대좌표에 영향을 끼치지 않는다.
  - ▶ 가지처럼 뻗어나가는 그림
- ▶ ‘+’ 2개: 구한 좌표를 뒤의 상대좌표 계산에 사용
  - ▶ 계속 이어지는 경로



## 텍스트 삽입 - 상대적 위치

text-rel.tex

점 라벨링하기: 크게 2가지 방법이 있다. (17.5절 참조)

```
1 \fill (0,0) circle [radius=2pt] node[above right] {A};
2 \fill (1,0) circle [radius=2pt] +(0.25,0.25) node {B};
```

$$\begin{array}{ccccc} \text{above left} & & \text{above} & & \text{above right} \\ & & \cdot & & \cdot \\ & & & & \\ & \text{left} & & & \text{right} \\ & \cdot & & & \cdot \\ & & & & \\ \text{below left} & & \text{below} & & \text{below right} \\ & \cdot & \cdot & & \cdot \end{array}$$

▶ `\usetikzlibrary{calc}`

```
1 | \draw (0,0) -- (3,1) -- ($ (3,1) + (-1,1) $);  
2 | \draw (0,0) -- ($ 2*(1,2) $);
```

▶ 13.5절 참조

▶ 보다 다양한 수학 계산은 pgfmath 관련 명령 사용

▶  $x^y$ , floor( $x$ ), isprime( $x$ ), int( $x$ ), rand, ...

▶ 89.2, 89.3, 90장 참조

▶ 힘들면 LuaTeX 사용

▶ `\RequirePackage{luatex85}`

# node와 coordinate

coordinate.tex

점에 이름 짓기:

```
1 \node [옵션] (이름) 위치 {내용};  
2 \coordinate [옵션] (이름) 위치;
```

```
1 \node (A) at (1,1) {a};  
2 \node (B) at (2,2) {b};  
3 \draw (A) -- (B);  
4 % \usetikzlibrary{positioning}  
5 \node (C) [below=of A] {c};
```

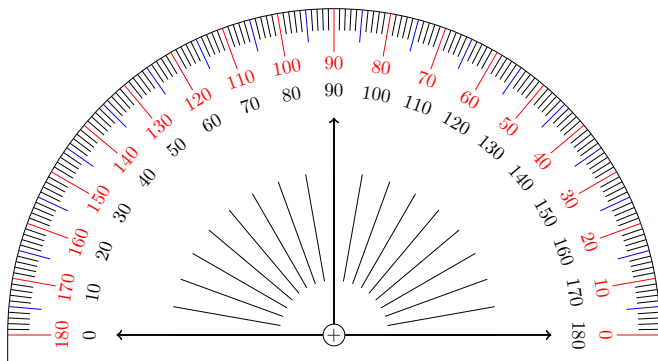
# 반복문의 활용

foreach.tex

- ▶ pgffor 패키지의 `\foreach`문은 평소에도 사용 가능.
- ▶ TikZ에서 잘 사용하면 강력한 도구가 된다.

```
1 % \usepackage{pgffor}
2 \draw (0,0) grid (5,5);
3 \foreach \i in {1,2,...,5}{
4     \draw (\i,\i) circle [radius=0.1*\i]
5     node [below right] {$ (x_{\i}, y_{\i}) $};
6 }
```

## 예시: 각도기 그리기



- ▶ `ifthen` 의 `\ifthenelse` 문, `intcalc` 의 `\intcalcMod` 사용.
- ▶ 소스: [git.io/vFrVL](https://git.io/vFrVL)

# scope 환경

scope.tex

- ▶ 원래 tikzpicture 환경에서 전역 옵션 사용가능.
  - ▶ `\begin{tikzpicture}[...]`, 혹은 `tikzset/tikzstyle`
- ▶ 이를 국소적으로 사용 가능케 한 것이 scope 환경.

```
1 \begin{tikzpicture}[blue]
2   \draw (0,0) -- (1,1);
3   \begin{scope}[red]
4     \draw (1,0) -- (2,1);
5     \draw (2,0) -- (3,1);
6   \end{scope}
7   \draw (3,0) -- (4,1);
8 \end{tikzpicture}
```

# clip

clip1.tex

- ▶ 기본적으로는 `\draw`와 동일.
- ▶ 그려진 도형 내부의 객체들만 그려짐

```
1  % draw option: draw a margin.  
2  \clip[draw] (0,0) circle [radius=2.6];  
3  \draw (-3,-3) grid (1,1);  
4  \draw (1,1.5) circle [radius=1];  
5  \fill (4,0) circle [radius=2.6];
```

## scope + clip

clip2.tex

- ▶ scope를 사용하지 않으면 clip은 전체에 적용됨.
- ▶ scope를 통해, 하나의 tikzpicture에 여러 그림 작도 가능

```
1 \begin{scope}
2     \clip (-2,0) circle [radius=3];
3     \fill[red] (-2,0) circle [radius=3];
4 \end{scope}
5 \begin{scope}
6     \clip (2,0) circle [radius=3];
7     \fill[blue] (2,0) circle [radius=3];
8     \fill[green] (-2,0) circle [radius=3];
9 \end{scope}
```



## 기타 팁

- ▶ 컴파일이 끝나질 않아요
  - ▶ 세미콜론(;)을 빠트렸을 가능성이 높습니다.
  - ▶ 보통 에러 메시지로 알려주는데, 가끔 무한루프를 도는 듯...
- ▶ 미리보기에서 선이 안 보여요
  - ▶ 선이 수평/수직일 경우 '화면'에서 보이지 않는 경우도.
- ▶ `\documentclass{standalone}`의 유용성
  - ▶ 그림이 본문 속에 작게 보이면 그리기 힘들.
  - ▶ 문서 전체를 조판할 때마다 그림을 새로 그리는 것은 낭비.
  - ▶ `pdfcrop`, `convert(ImageMagick)` 등의 후처리가 편해짐

## 이번 시간에 안 한 것들

- ▶ 그라데이션(Shading) / 색 배합(xcolor)
- ▶ `\tikzset`, `\tikzstyle`의 활용
- ▶ TikZ 관련 패키지의 활용
- ▶ 애니메이션 만들기
- ▶ GeoGebra에서의 TikZ
- ▶ 기타; 수없이 많은 명령어의 수없이 많은 옵션
- ▶ 등등...

# TikZ의 왕도

- ▶ 기초 문법에 익숙해지기
  - ▶ 여기저기서 최대한 많이 써보기 // 쓸 일이 많아...
- ▶ 간결하게 코딩하려 노력하기
  - ▶ TikZ의 이점을 살리는 길.
  - ▶ 같은 결과라도 더 간단하고 직관적인 방법이 있을수도!
- ▶ 남의 코드에서 배우기
  - ▶ 심심할 때마다 `pgfmanual`, `texample.net`
  - ▶ 오늘 워크숍을 열심히 보고 듣기...

# TikZ의 장단점

(사용자 입장에서.)

- ▶ 장점:

- ▶ 무료!
- ▶ 반복작업 가능 / 다양한 패키지
- ▶ 단순하지만 품위 있는 그림을 빠르게...
- ▶ **표현 수단의 확대 → 표현의 확대 → 생각의 확대**

- ▶ 단점:

- ▶ 좌표 설정이 까다로움
- ▶ 너무 긴 분량의 매뉴얼
- ▶ 과거의 PSTricks에 비하여 빈약한 생태계

# 읽을거리

- ▶ [pgfmanual](#): TikZ/PGF의 바이블. 너무 길다는 단점.
- ▶ [VisualTikZ](#): 시각적인 사용 가이드. (VisualPSTricks에서 유래)
- ▶ [awesome-tikz](#): TikZ 관련 읽을거리 모음
  - ▶ 그 중에서도:
    - ▶ [minimaltikz](#): 급한 사람들을 위하여:
    - ▶ [TeXample.net](#): 다양한 TikZ 예시 수록
    - ▶ [Nice Scientific Pictures](#): 걸작 모음
- ▶ [seungwonpark/tikz-gallery](#): 본 발표자의 TikZ 작품(?) 모음
- ▶ 그냥, 심심하다면(?)...
  - ▶ TikZ의 저자, Till Tantau의 2015 dante 컨퍼런스 발표자료:  
[Small TikZ miracles](#) (독일어)

## 참고 문헌

- ▶ T.Tantau (2015), The TikZ and PGF Packages (`texdoc tikz`)
- ▶ 윤석천 (2015), LaTeX 쉽게 쓰기, [bit.ly/2tXsRgi](http://bit.ly/2tXsRgi)
- ▶ J.P.Casteleyn (2017), Visual TikZ (`texdoc visualtikz`)
- ▶ 박승원 (2017), Graphing/Drawing Tools for Physics Research, [goo.gl/Lz2gBL](http://goo.gl/Lz2gBL)
- ▶ 조인성 (2017), 개인 서신

감사합니다