

L^AT_EX: from dummy to T_EXnician

The bases of TikZ

Anton Lioznov

Skoltech

ISP 2019,
lesson 3

What we will know?

Introduction: on approaches to TikZ

“General usage”

“Graphs”

“Arrangement”

Acknowledgments

We acknowledge

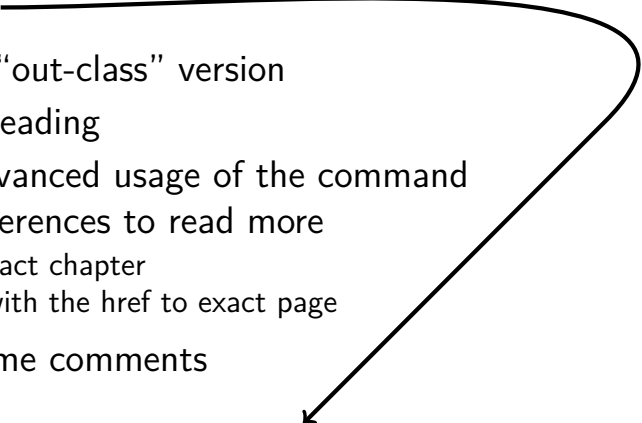
Dmitry Barashev for useful comments, that was included in the presentation

Alexander Kulikov for useful comments, that was included in the presentation

Agreements

I

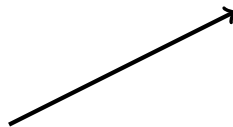
Footnotes

- ▶ Only in the “out-class” version
 - ▶ For second reading
 - ▶ Containe advanced usage of the command
 - ▶ Containe references to read more
 - ▶ to the exact chapter
 - ▶ (often) with the href to exact page
 - ▶ Containe some comments
- 

Like this



Addition information – “magic”



- ▶ To have the full picture
- ▶ Not to analyze or to puzzle out in class

What we have learned today?

Introduction: on approaches to TikZ

“General usage”

Coordinates

“Graphs”

Common graphs

Trees

“Arrangement”

What we will know?

Introduction: on approaches to TikZ

“General usage”

“Graphs”

“Arrangement”

What is TikZ?

“TikZ ist kein Zeichenprogramm”

which translates to “TikZ is not a drawing program”

TikZ defines a number of \TeX commands that produce graphics: 
produced by `\tikz \fill[orange] (1ex,1ex) circle (1ex);`

Pros and Cons

Cons

- it is most likely that you don't need TikZ
- write visual-based things like graphics is really annoying in a not-WYSiWYG way

Pros

- + it is most likely that you need some TikZ elements
- + some graphics (graphs for example) are so good structured, that it is OK to program them
- + TikZ has perfect integration with \LaTeX (and beamer):
 - ▶ You can use all \LaTeX commands inside TikZ, creating beautiful pictures with math
 - ▶ You can pose elements using TikZ
 - ▶ You can show just part of the picture in beamer Overlays
- + You don't need to have an external file
- + TikZ is using in CV and lots of other templates. It is good to be able to read the code

How to setup TikZ picture?

`\usepackage{tikz}`

and then

`\begin{tikzpicture} <code> \end{tikzpicture}` or, for short inline graphics, `\tikz`.

What we will know?

Introduction: on approaches to TikZ

“General usage”

“Graphs”

“Arrangement”

Path

Path is the main TikZ essence.

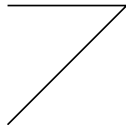
```
\usepackage{tikz}
\begin{tikzpicture}
\path (0,0) -- (1,1) -- (0, 1);
\end{tikzpicture}
```

“(0,0)”, “(0,1)” is the simplest coordinate assignment. The (x, y) coordinate in units (typically 1cm)

The `\path` is not draw anything by itself!

Draw

```
\usepackage{tikz}
\begin{tikzpicture}
\path[draw] (0,0) -- (1,1) -- (0, 1);
\end{tikzpicture}
```



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

Use `\draw` or `\path[draw]` to draw

Fill

```
\usepackage{tikz}
\begin{tikzpicture}
\fill (0,0) -- (1,1) -- (0, 1) --
    cycle;
\end{tikzpicture}
```



Use `\fill` or `\path[fill]` to fill.

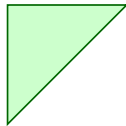
And `--cycle` to make the path close

Options

Let us add some color!

```
\usepackage{xcolor}

\begin{tikzpicture}
\filldraw[fill=green!20!white, draw=
  green!40!black] (0,0) -- (1,1) --
  (0, 1) -- cycle;
\end{tikzpicture}
```



new command `\filldraw`.

Notice the *key=value* syntax!

Options

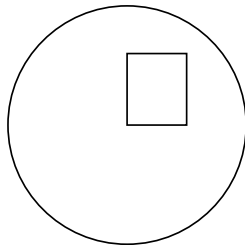
Draw another line

```
\usepackage{tikz}  
\begin{tikzpicture}  
\draw[->, very thick, dotted] (0,0)  
  -- (1,1) -- (0, 1);  
\end{tikzpicture}
```



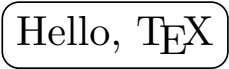
Other figures

```
\usepackage{tikz}  
\begin{tikzpicture}  
\draw (0,0) circle [radius=1cm];  
\draw (0,0) rectangle (0.5, 0.6);  
\end{tikzpicture}
```



Nodes

```
\usepackage{tikz}
\begin{tikzpicture}
\node[draw, rounded corners] at (0,0)
    {Hello, \TeX};
\end{tikzpicture}
```

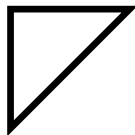


Hello, T_EX

`\node` or `\path[node]`. Without optional arguments a node has no border

Vertical and horizontal

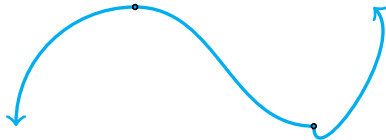
```
\usepackage{tikz}  
\begin{tikzpicture}[ultra thick]  
\draw (0,0) -- (1,1) -| cycle;  
\end{tikzpicture}
```



Use `-|` for “first horizontal, then vertical”. Use `|-` for “first vertical, then horizontal”

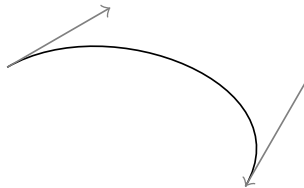
Curves

```
\usepackage{tikz}
\begin{tikzpicture}
\draw [<->,thick,cyan] (0,0) to [out
    =90,in=180] (1,1)
to [out=0,in=180] (2.5,0) to [out
    =-90,in=-45] (3,1) ;
\draw (1,1) circle(0.02cm);
\draw (2.5,0) circle(0.02cm);
\end{tikzpicture}
```



to [out=.., in=..] the angle on which curve flows out and the angle on which curve flows in.

```
\usepackage{tikz}
\begin{tikzpicture}
\draw (1,0) .. controls +(30:1cm) and
    +(60:1cm) .. (3,-1);
\draw[gray,->] (1,0) -- +(30:1cm);
\draw[gray,<-] (3,-1) -- +(60:1cm);
\end{tikzpicture}
```



.. controls <coord> and <coord> ..

What we will know?

“General usage”
Coordinates

“standart”

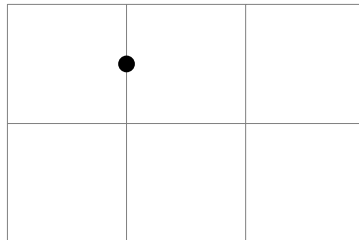
x,y

x, y:

```
\begin{tikzpicture}  
  \draw[help lines] (0,0) grid (3,2);  
  \fill (1cm,1.5cm) circle (2pt);  
  \fill (2cm,-5mm+2pt) circle (2pt);  
\end{tikzpicture}
```

(<x-coord>, <y-coord>)

x, y:



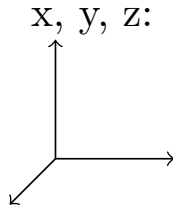
“standart”

x,y,z

x, y, z:

```
\begin{tikzpicture}[->]  
\draw (0,0) -- (1,0);  
\draw (0,0) -- (0,1,0);  
\draw (0,0) -- (0,0,1);  
\end{tikzpicture}
```

(<x-coord>, <y-coord>, <z-coord>)



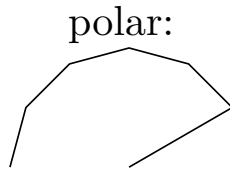
“standart”

θ, r

polar:

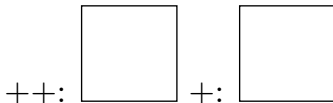
```
\begin{tikzpicture}  
\draw (0cm,0cm) -- (30:1cm) -- (60:1  
cm) -- (90:1cm)  
-- (120:1cm) -- (150:1cm) -- (180:1cm  
);  
\end{tikzpicture}
```

($\langle\theta\text{-coord}\rangle:\langle r\text{-coord}\rangle$)



“++” and “+=+” coordinates

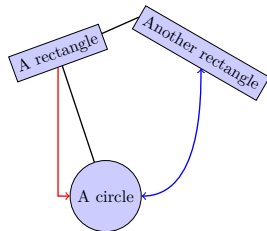
```
++:  
\begin{tikzpicture}  
\draw (2,0) -- ++(1,0) -- ++(0,1) --  
      ++(-1,0) -- cycle;  
\end{tikzpicture}  
+:  
\begin{tikzpicture}  
\draw (2,0) -- +(1,0) -- +(1,1) --  
      +(0,1) -- cycle;  
\end{tikzpicture}
```



- ▶ “++” use relative coordinate and set this new coordinate as “current”
- ▶ “+” use relative coordinate and DOESN'T set this new coordinate as “current”

“Node”-based

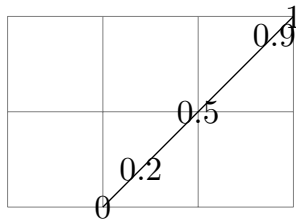
```
\path (3,-1) node(b) [circle,draw,
    fill] {A circle}
(2,2) node(c) [rectangle,rotate=20,
    draw,fill] {A rectangle}
(5,2) node(d) [rectangle,rotate=-30,
    draw,fill] {Another rectangle};
\draw[thick] (b) -- (c) -- (d.west);
\draw[thick,red,->] (c) |- (b);
\draw[thick,blue,<->] (b) .. controls
    +(right:2cm) and +(down:1cm) ..
    (d);
\end{tikzpicture}
```



1. label node | (`<label>`)
2. refer to the node as (`node cs:name=<label>`)
3. you can also use things like `<label>.west` or `<label>.right`



```
\usepackage{tikz}
\usetikzlibrary{calc}
\begin{tikzpicture}
\draw [help lines] (0,0) grid (3,2);
\draw (1,0) -- (3,2);
\foreach \i in {0,0.2,0.5,0.9,1}
\node at ($(1,0)!\i!(3,2)$) {\i};
\end{tikzpicture}
```

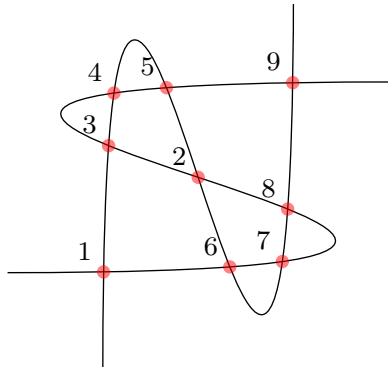


1. `\usetikzlibrary{calc}`
2. syntax: `$<coord1>!\fraction!<coord2>$`
3. in this slide you can also see `\foreach`!

Coordinate intersection



```
\usepackage{tikz}
\usetikzlibrary{intersections}
\begin{tikzpicture}
\clip (-2,-2) rectangle (2,2);
\draw [name path=curve 1] (-2,-1) ..
    controls (8,-1) and (-8,1) ..
    (2,1);
\draw [name path=curve 2] (-1,-2) ..
    controls (-1,8) and (1,-8) ..
    (1,2);
\fill [name intersections={of=curve 1
    and curve 2, name=i, total=\t}]
[red, opacity=0.5, every node/.style
    ={above left, black, opacity=1}]
\foreach \s in {1,...,\t}{(i-\s)
    circle (2pt) node {\footnotesize\
    s}};
\end{tikzpicture}
```



What we will know?

Introduction: on approaches to TikZ

“General usage”

“Graphs”

“Arrangement”

What we will know?

“Graphs”

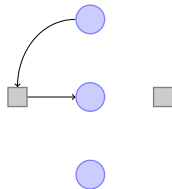
Common graphs

Trees

Graph example 1

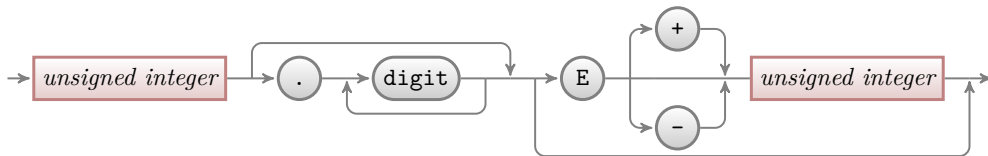


```
\usepackage{tikz}
\usetikzlibrary{positioning}
\begin{tikzpicture}[place/.style={circle,draw=
    blue!50,fill=blue!20,thick,
    inner sep=0pt,minimum size=6mm},
    transition/.style={rectangle,draw=black!50,fill=
    black!20,thick,
    inner sep=0pt,minimum size=4mm}]
\node[place] (waiting) {};
\node[place] (critical) [below=of waiting] {};
\node[place] (semaphore) [below=of critical] {};
\node[transition] (leave critical) [right=of
    critical] {};
\node[transition] (enter critical) [left=of
    critical] {};
\draw [->] (enter critical) to (critical);
\draw [->] (waiting) to [out=180,in=90] (enter
    critical);
\end{tikzpicture}
```



You can write `below=of <label>` to have a relative coordinate

Chain example



What we will know?

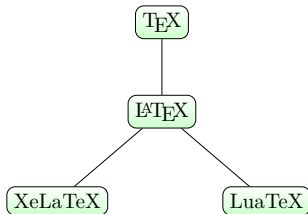
“Graphs”

Common graphs

Trees

Tree

```
\usepackage{tikz}
\begin{tikzpicture}[sibling distance=10em,
  every node/.style = {shape=rectangle, rounded
    corners,
    draw, align=center,
    top color=white, bottom color=green!20}]
\node {\TeX}
  child { node {\LaTeX}
    child { node {\XeLaTeX} }
    child { node {\LuaTeX} }
  } ;
\end{tikzpicture}
```



We use `\node` and `child`.

`sibling distance` option provides a horizontal distance between nodes

What we will know?

Introduction: on approaches to TikZ

“General usage”

“Graphs”

“Arrangement”



TikZ is often used not as “independent picture”, but as a part of the presentation or document.

Examples: This arrow was used to show you, where the “magic page” indicator is located

```
\smash{\begin{tikzpicture}  
\draw[white] (0,0) -- (0, 0.6);  
\draw[->,ultra thick] (0,0.3) --(3,1.8);  
\end{tikzpicture}}
```

the “in class task”



This was produced by

```
\tikz[baseline]\node[anchor=base,draw=red,rounded corners,inner xsep=1ex,inner  
ysep=0.75ex,bottom color=red!20,top color=red!10]{In class task};
```



\LaTeX is only for use
in academic area

“Common belief”



L^AT_EX is common use
in academic area

WRONG

“Common belief”



L^AT_EX is only for use
in academic area




WRONG

was produce by

```
\begin{tikzpicture}
\node[align=center] (0,0) {
\huge \LaTeX\ is only for use\\ \huge in academic area
};
\uncover<2,3>{\node[rotate=30, bottom color=red!50, top color=red!50] (0,0) {\Huge
WRONG}};
\end{tikzpicture}
```

references I

color from the footnotes corresponds to references' color.

- ▶  : <https://tex.stackexchange.com/questions>
- ▶  : <https://en.wikibooks.org/wiki/LaTeX>
- ▶  : <https://www.overleaf.com/learn/latex>
- ▶ **TikZ: TikZ manual** (1161 pages :))

Distribution

- ▶ the pdf-version of the presentation and all printed materials can be distributed under license Creative Commons Attribution-ShareAlike 4.0
<https://creativecommons.org/licenses/by-sa/4.0/>
- ▶ The source code of the presentation is available on <https://github.com/Lavton/latexLectures> and can be distributed under the MIT license
https://en.wikipedia.org/wiki/MIT_License#License_terms