LaTeX: from dummy to TeXnician

TikZ and Typography

Anton Lioznov, Anna Litvin

Skoltech

ISP 2025,
lesson 3

What we will know?

Technical agreements

TikZ

(User-level) Typography

What we will know?

Technical agreements

TikZ

(User-level) Typography

Agreements

inclass/outclass versions

- two slightly different versions for class and home
- class version is more interactive and contains less information
- ⟷ this line will be shown only at home version

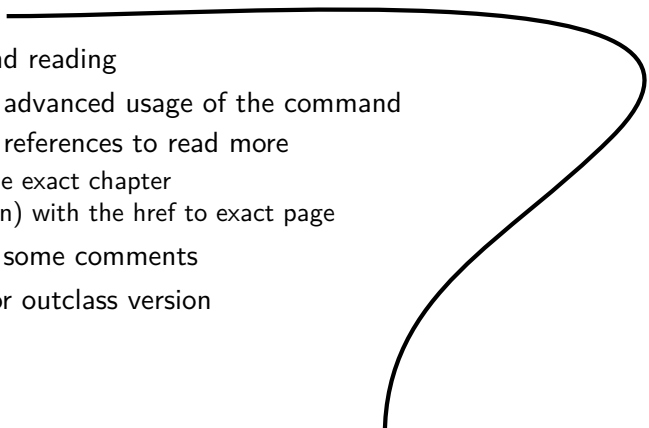
Frame for home

Agreements

II

Footnotes

- For second reading
- Contains advanced usage of the command
- Contains references to read more
  - to the exact chapter
  - (often) with the href to exact page
- Contains some comments
- Mostly for outclass version



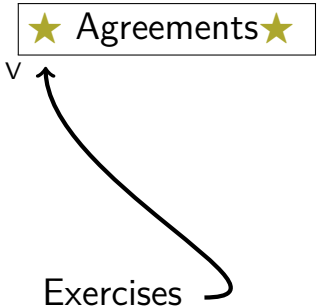
Like this

Agreements

III

Addition information – “magic”

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



- To work in class

Special thanks to

- Anna Pavlicheva for counseling of non-techniqual part about fonts

Our TAs:

- Peter Borisovets
- Pavel Kuzmin
- Anna Litvin

What we will know?

Technical agreements

TikZ

(User-level) Typography

What we will know?

TikZ

- Introduction: what is TikZ and when to use it
- General usage
- Graphs
- Arrangment

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 thinks 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.

```
\usepackage{color}
\usepackage{xcolor}
\usepackage{tikz}

\begin{document}
    \begin{tikzpicture}
        \fill[green] (1ex,1ex) circle (1ex);
    \end{tikzpicture}
    \tikz \fill[orange] (1ex,1ex) circle (1ex);
\end{document}
```

Examples

- https://tikz.net/
- https://texample.net/bar-chart/

What we will know?

TikZ

- Introduction: what is TikZ and when to use it
- General usage
- Graphs
- Arrangment

Path

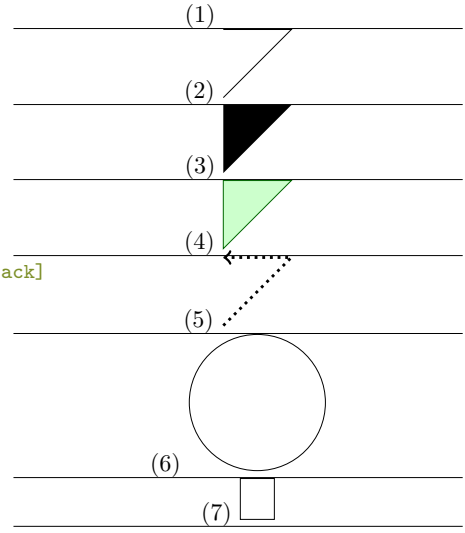
Path is the main TikZ essence.

```
\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}
\hrule
(1) \tikz \path (0,0) -- (1,1) -- (0, 1);
↪  \par \hrule
(2) \tikz \draw (0,0) -- (1,1) -- (0, 1);
↪  \par \hrule
(3) \tikz \fill (0,0) -- (1,1) -- (0, 1)
↪  -- cycle; \par \hrule
(4) \tikz
↪  \filldraw[fill=green!20!white,draw=green!40!black]
↪  (0,0) -- (1,1) -- (0, 1) -- cycle;
↪  \par\hrule
(5) \tikz\draw[->, very thick, dotted]
↪  (0,0) -- (1,1) -- (0, 1);\par\hrule
(6) \tikz \draw (0,0) circle
↪  [radius=1cm];\par\hrule
(7) \tikz \draw (0,0) rectangle (0.5,
↪  0.6);\par\hrule
```



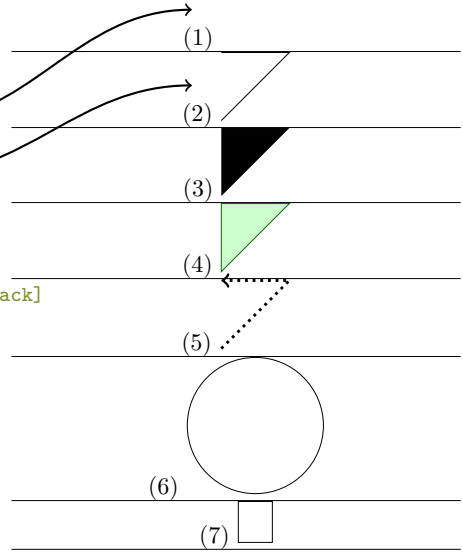
\draw ≡ \path[draw],

\fill ≡ \path[fill]

etc

Draw

```
\usepackage{tikz}
\hrule
(1) \tikz \path (0,0) -- (1,1) -- (0, 1);
↪  \par \hrule
(2) \tikz \draw (0,0) -- (1,1) -- (0, 1);
↪  \par \hrule
(3) \tikz \fill (0,0) -- (1,1) -- (0, 1)
↪  -- cycle; \par \hrule
(4) \tikz
↪  \filldraw[fill=green!20!white,draw=green!40!black]
↪  (0,0) -- (1,1) -- (0, 1) -- cycle;
↪  \par\hrule
(5) \tikz\draw[->, very thick, dotted]
↪  (0,0) -- (1,1) -- (0, 1);\par\hrule
(6) \tikz \draw (0,0) circle
↪  [radius=1cm];\par\hrule
(7) \tikz \draw (0,0) rectangle (0.5,
↪  0.6);\par\hrule
```



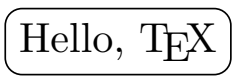
\draw ≡ \path[draw],

\fill ≡ \path[fill]

etc

Nodes

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

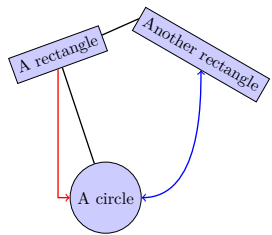


`\node` or `\path[node]`. Without optional arguments a node has no border.

Nodes are important for graph creation

“Node”-based coordinates

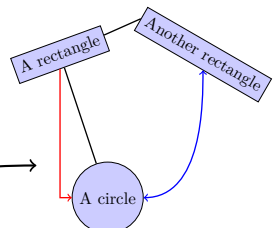
```
\begin{tikzpicture}[fill=blue!20]
    \path (3,-1) node(b)
    ↪  [circle,draw,fill] {A circle}
    (2,2) node(c)
    ↪  [rectangle,rotate=20,draw,fill] {A
    ↪  rectangle}
    (5,2) node(d)
    ↪  [rectangle,rotate=-30,draw,fill]
    ↪  {Another rectangle};
    \draw[thick] (b) -- (c) -- (d.west);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



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

“Node”-based coordinates

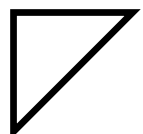
```
\begin{tikzpicture}[fill=blue!20]
    \path (3,-1) node(b)
    ↪  [circle,draw,fill] {A circle}
    (2,2) node(c)
    ↪  [rectangle,rotate=20,draw,fill] {A
    ↪  rectangle}
    (5,2) node(d)
    ↪  [rectangle,rotate=-30,draw,fill]
    ↪  {Another rectangle};
    \draw[thick] (b) -- (c) -- (d.west);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



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

Vertical and horizontal

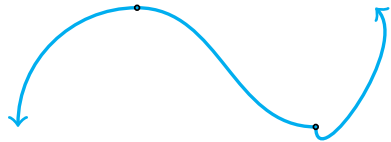
```
\usepackage{tikz}
\begin{tikzpicture}[ultra thick]
    \draw (0,0) -- (1,1) -| cycle;
\end{tikzpicture}
```



Use -| for “first horizontal, than vertical”. Use |- for “first vertical, than horizontal”

Curves

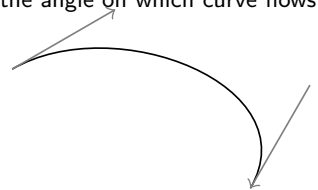
```
\usepackage{tikz}

\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> ..`

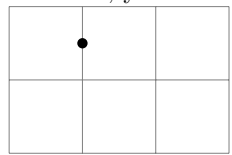


https://en.wikipedia.org/wiki/Bezier_curve

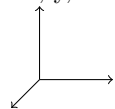
Coordinates

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

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

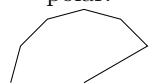
x, y:



x, y, z:

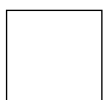
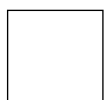


polar:



“++” and “+” coordinates

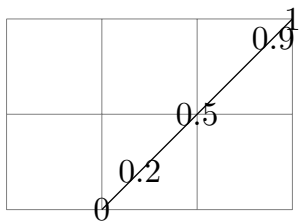
```
++:
\begin{tikzpicture}
    \draw (2,0) -- ++(1,0) -- ++(0,1) --
    ↪  ++(-1,0) -- cycle;
\end{tikzpicture}
+:
\begin{tikzpicture}
    \draw (2,0) -- +(1,0) -- +(1,1) --
    ↪  +(0,1) -- cycle;
\end{tikzpicture}
```

++:  +: 

- “++” use relative coordinate and set this new coordinate as “current”
- “+” use relative coordinate and DOESN’t set this new coordinate as “current”

Coordinate calculation

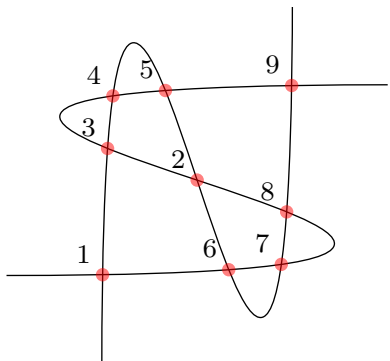
```
\usepackage{tikz}
\usetikzlibrary{calc}

\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?

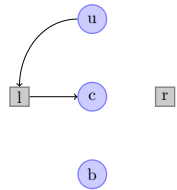
TikZ

- Introduction: what is TikZ and when to use it
- General usage
- Graphs
- Arrangment

TikZ allows graphics to work according to “rules”. This is a minus for an arbitrary drawing, but it is an advantage for those drawings that have a given structure and are also built according to “rules”

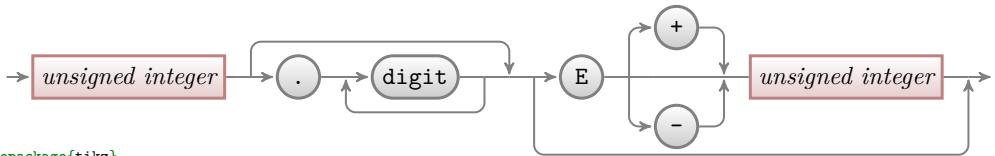
Graph example 1

```
\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] (Node u) {u};
    \node[place] (my center) [below=of
    ↪  Node u] {c};
    \node[place] (Node b) [below=of my
    ↪  center] {b};
    \node[transition] (Node r) [right=of
    ↪  my center] {r};
    \node[transition] (Node l) [left=of my
    ↪  center] {l};
    \draw [->] (Node l) to (my center);
    \draw [->] (Node u) to [out=180,in=90]
    ↪  (Node l);
\end{tikzpicture}
```



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

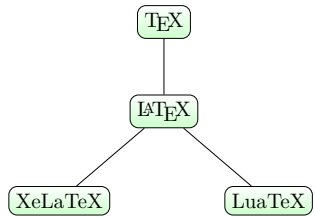
Chain example




```
\usepackage{tikz}
\usetikzlibrary{graphs, arrows, shapes.misc}
\begin{tikzpicture}[>=stealth', black!50, text=black, thick, every new ->/.style = {shorten >=1pt},
graphs/every graph/.style = {edges=rounded corners}, skip loop/.style = {to path={-- ++(0,#1) -| (\tikztotarget)}}, hv
↪  path/.style = {to path={-| (\tikztotarget)}}, vh path/.style = {to path={|- (\tikztotarget)}},
nonterminal/.style = {rectangle, minimum size=6mm, very thick, draw=red!50!black!50, top color=white, bottom
↪  color=red!50!black!20, font=\itshape, text height=1.5ex,text depth=.25ex},
terminal/.style = {rounded rectangle, minimum size=6mm, very thick, draw=black!50, top color=white,bottom color=black!20,
↪  font=\ttfamily, text height=1.5ex, text depth=.25ex},shape = coordinate]
    \graph [grow right sep, branch down=7mm, simple] {
        / -> unsigned integer[nonterminal] -- p1 -> "." [terminal] -- p2 -> digit[terminal] --
        p3 -- p4 -- p5 -> E[terminal] -- q1 ->[vh path]
        {[nodes={yshift=7mm}]
        "+"[terminal], q2, "-"[terminal]
        } -> [hv path]
        q3 -- /unsigned integer [nonterminal] -- p6 -> /;
        p1 ->[skip loop=5mm] p4;
        p3 ->[skip loop=-5mm] p2;
        p5 ->[skip loop=-11mm] p6;
        q1 -- q2 -- q3; % make these edges plain
    };
\end{tikzpicture}
```

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?

TikZ

- Introduction: what is TikZ and when to use it
- General usage
- Graphs
- Arrangment

Very few people make full-fledged charts in tikZ. But it is much more common to add tikZ as text decoration element.

Introduction

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

Example: CV

```
\newcommand{\cvtag}[1]{%
    \tikz[baseline]\node[anchor=base,
    ↪  draw=body!30, rounded corners,
    ↪  inner xsep=1ex, inner ysep
    ↪  =0.75ex, text height=1.5ex, text
    ↪  depth=.25ex]{#1};
}

I can use:\\ \cvtag{Python}
↪  \cvtag{\LaTeX}\\
```

I can use:

Python LaTeX

Also I have some experience in:
Optimization , Modeling ,

Example: CV

```
\newcommand{\sectorskill}[1] {%
  \begin{tikzpicture}[scale=0.3]
  \pgfmathtruncatemacro\fillsector{#1 *
  ↪  360 / 10}
  \filldraw[fill=accent!10!white,
  ↪  draw=accent!50!black, rotate=90]
    (0,0) -- (3mm,0mm) arc (0:360:3mm) --
    ↪  (0,0);
  \filldraw[fill=accent!40!white,
  ↪  draw=accent!50!black, rotate=90]
    (0,0) -- (3mm,0mm) arc
    ↪  (0:\fillsector:3mm) -- (0,0);
  \end{tikzpicture}
}

Also I have some experience in:\\
↪  Optimization\sectorskill{2},
↪  Modeling\sectorskill{7},
```

I can use:

Python LaTeX

Also I have some experience in: Optimization  , Modeling ,

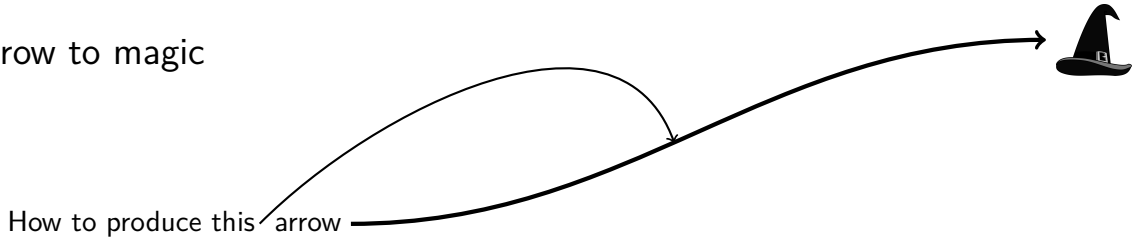
“Magic”

How to produce “magic”

```
\begin{tikzpicture}[remember picture,overlay,shift={(current page.north east)}]
\node[anchor=north east,xshift=-0cm,yshift=-0cm](endM) {%
    {\includegraphics[width=1cm]{images/magic}}%
};
\end{tikzpicture}
```

Notice `shift={(current page.north east)}`

arrow to magic




```
% remember position of (startM) node (and (endM) node from previous slide)
How to produce this arrow \tikz[overlay, remember picture] \node (startM) {};

% go from (startM) to (endM)
\tikz[overlay, remember picture] \draw[->, ultra thick] ($(startM)+(0,0.1)$) to[out=0,in=180]
↪  (endM);
```


What about arrow pointed on the thick arrow? It was produced with intersections library

“Common belief”

LaTeX is only for use
in academic area

“Common belief”

“Common belief”

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}
```

What we will know?

Technical agreements

TikZ

(User-level) Typography

What we will know?

- (User-level) Typography
  - Text position
  - Other
  - Fonts

textpos

```
%add `showboxes` to view the block
\usepackage[absolute, overlay]{textpos}

\begin{textblock}{2}(1, 1.2)
\noindent
Y e l l o
\end{textblock}

\begin{textblock*}{10mm}(10.5mm,10.5mm)
\noindent
H e l l o
\end{textblock*}
```

Y e l l
o H e l l
o

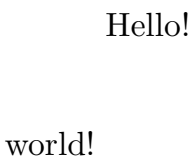
http://mirrors.ctan.org/macros/latex/contrib/textpos/textpos.pdf

Textpos: man

we have
`{textblock}{block_height}(position_x, position_y)`
in `textblock*` all values are in length units.
in `textblock` all values are in units of mesh that is stretches on the page. See `\TPoptions`, `\TPGrid` in manual for more

Absolute position with TikZ

```
\usepackage{tikz}
\begin{tikzpicture}[remember
↪  picture,overlay,shift={(3cm, -1cm)}]
\node[anchor=north
↪  ,xshift=-0cm,yshift=0.1cm]{\noindent
↪  Hello!};
\end{tikzpicture}

\begin{tikzpicture}[remember
↪  picture,overlay,shift={(current
↪  page.center)}]
\node[anchor=north
↪  ,xshift=-0cm,yshift=0.0cm]{\noindent
↪  world!};
\end{tikzpicture}
```



- `remember picture` to reference outside the current position
- `shift` to shift

What we will know?

- (User-level) Typography
  - Text position
  - Other
  - Fonts

Rules


```
\rule{3em}{2em}\\

\hrule width6cm\relax~\\
\hrule\relax~\\

\vrule height1cm depth2pt\relax
```

`\rule` provide rectangle `\hrule` and `\vrule` provides a horizontal and veritcal lines. The keywords for them are `width`, `height`, `depth` and all can be ommited

kn: 21

In these lectures we use rules in frame style and to show length and box.

Leaders

Leaders are generic for glue. They are using in tables of content.

Like . . . . . this

```
\leaders\hbox to 1em{\hss.\hss}\hfill
```

Geometry

```
\usepackage{geometry}
\geometry{a4paper,paperwidth=70mm, paperheight=65mm, left=5mm, top=5
mm, right=5mm, bottom=5mm, layoutwidth=60mm, layoutheight=35mm}
```

change the shape of the paper as you like

Lorem ipsum

```
\usepackage{lipsum}

\lipsum[1]
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

http://mirrors.ctan.org/macros/latex/contrib/lipsum/lipsum.pdf

What we will know?

- (User-level) Typography
  - Text position
  - Other
  - Fonts

Fonts classification

- Serif — for long texts, books,..

  The quick brown fox jumps over the sleazy dog

- Sans Serif — for short texts, titles, presentations,..

  The quick brown fox jumps over the sleazy dog

- Typewriter — emulate typewriter, write code and commands

  The quick brown fox jumps over the sleazy dog

- Other — Decoration etc

  The quick brown fox jumps over the sleazy dog

http://www.tug.dk/FontCatalogue/accanthis/ http://www.tug.dk/FontCatalogue/arev/
http://www.tug.dk/FontCatalogue/ascii/ http://www.tug.dk/FontCatalogue/janaskrivana/

TEXnical classification

you have:

standard pdfLATEX engine with “METAFONT” fonts:

The package has **global** usage out-of-the-box you want to use it **globally**	The package has **only global** usage out-of-the-box you want to use it **locally**
The package has **only local** usage out-of-the-box you want to use it **globally**	The package has **local** usage out-of-the-box you want to use it **locally**

XELATEX with support of system-installed fonts:

The font is **global** you want to use it **globally**	The font is **global** you want to use in **locally**
The font is **local** you want to use in **globally**	tThe font is **local** you want to use in **locally**

pdfLaTeX

Global Font usage throw package with pdfLaTeX

where to find a font

- https://tug.org/FontCatalogue/allfonts.html
- https://www.ctan.org/tex-archive/fonts

then just follow the instructions for the package

Fonts usage

notation

You may have something like this in logs:

`LaTeX Font Warning: Font shape `T1/calligra/bx/n' undefined`

Or something like `OT1/cmr/m/n/10`

How to read it?

T1	calligra	bx	n	
OT1	cmr	m	it	10
encoding	font family	series	shape	font size

http://mirrors.ctan.org/macros/latex/doc/fntguide.pdf: 2.1, http://mirrors.ctan.org/macros/latex/doc/encguide.pdf

Fonts usage

notation

Most common encodings	
OT1	TEX text
T1	TEX extended text
OML	TEX math italic
OMS	TEX math symbols
OMX	TEX math large symbols
U	Unknown
L<xx>	local encoding

Some common families	
cmr	Computer Modern Roman
cmss	Computer Modern Sans
cmtt	Computer Modern Typewriter
cmm	Computer Modern Math Italic
cmsy	Computer Modern Math Symbols
cmex	Computer Modern Math Extensions
ptm	Adobe Times
phv	Adobe Helvetica
pcr	Adobe Courier

more families: /Font_typefaces http://mirrors.ctan.org/macros/latex/doc/fntguide.pdf: 2.1

Look at the source code to understand how the right table was created (`>{\footnotesize}` and `\usepackage{array}`)

Fonts usage

notation

Most common values for series	
t	thin
m	Medium
b	Bold
bx	Bold extended
sb	Semi-bold
c	Condensed

Most common values for shape	
n	Normal (that is 'upright' or 'roman')
it	Italic
sl	Slanted (or 'oblique')
sc	Caps and small caps

http://mirrors.ctan.org/macros/latex/doc/fntguide.pdf: 2.1

Fonts usage

default

```
% 01
% pdfLaTeX, default
\usepackage{fontenc}
\usepackage[utf8]{inputenc}
{\LARGE fi} % ligature
the quick brown fox jumps over the lazy
↪ dog
\bfseries the quick brown fox jumps
↪ over the lazy dog
THE QUICK BROWN FOX JUMPS OVER THE LAZY
↪ DOG
0123456789
```

\usepackage{fontenc}

fi

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789

Global Font usage throw package with pdfLaTeX, when the package is constructed to change defaults

Fonts usage

Global font by loading package

```
\usepackage{venturisold}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
{\LARGE fi} % ligature
the quick brown fox jumps over the lazy
↪ dog
\bfseries the quick brown fox jumps
↪ over the lazy dog
THE QUICK BROWN FOX JUMPS OVER THE LAZY
↪ DOG
0123456789
```

```
\usepackage{<fontPackage>}
\usepackage[T1]{fontenc}
```

fi

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789

Fonts usage

Global font by loading package

```
\usepackage{berenis}
\usepackage[LY1]{fontenc}
\usepackage[utf8]{inputenc}
{\LARGE fi} % ligature
the quick brown fox jumps over the lazy
↪ dog
\bfseries the quick brown fox jumps
↪ over the lazy dog
THE QUICK BROWN FOX JUMPS OVER THE LAZY
↪ DOG
0123456789
```

```
\usepackage{<fontPackage>}
\usepackage[LY1]{fontenc}
```

fi

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789

Fonts usage

Global font by loading package

```
\usepackage[sfdefault,thin]{FiraSans}
↪  %% option 'sfdefault' activates
↪  Fira Sans as the default text font
\usepackage[T1]{fontenc}
\renewcommand*\oldstylenums[1]{{\firaoldstyle
↪  #1}}
\usepackage[utf8]{inputenc}

{\LARGE fi} % ligature

the quick brown fox jumps over the lazy
↪  dog

\bfseries the quick brown fox jumps
↪  over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY
↪  DOG

0123456789
```

fi

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789

Local font usage throw package with pdfLaTeX, when the package is constructed to use locally

Fonts usage

Local font by loading package

```
\usepackage{humanist}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
{\LARGE fi} (default)\\
\hminfamily {\LARGE fi} % ligature
```

fi (default)
fi

the quick brown fox jumps over
the quick brown fox jumps over
the lazy dog
the quick brown fox jumps
the quick brown fox jumps
over the lazy dog
THE QUICK BROWN FOX JUMPS
THE QUICK BROWN FOX JUMPS OVER
THE LAZY DOG
0123456789 0123456789

All that is done here and bellow is just follow
http://www.tug.dk/FontCatalogue/allfonts.html

Fonts usage

Local font by loading package

```
\input EileenBl.fd
\newcommand*\initfamily{\usefont{U}{EileenBl}{xl}{n}}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

{\LARGE fi} (default)\\
\initfamily{\LARGE fi} % ligature

\initfamily the quick brown fox jumps
↪  over the lazy dog
```

fi (default)

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

There are some beautiful fonts!

Fonts usage

Local font by loading package

```
\usepackage{calligra}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

{\LARGE fi} 0123456789 (default)\\
\calligra {\LARGE fi} % ligature

\calligra the quick brown fox jumps
↪  over the lazy dog

\calligra \bfseries the quick brown fox
↪  jumps over the lazy dog
```

fi 0123456789 (default)

fi 0123456789

the quick brown fox jumps over

the quick brown fox jumps over the lazy dog

the quick brown fox jumps

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

Global font usage throw package with pdfLaTeX, when the package is constructed to use locally

Algorithm

You need to figuraute the Font Family

1. Check the package documentation
2. (Remember, that not all fonts provide all series and shapes!)
3. If manual is unreachable, get the Family directly: `\showthe\font` and see logs:

```
\usepackage{humanist}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
{
\hminfamily \showthe\font
}
```

```
> \OT1/hmin/m/n/10 .
<recently read> \font

l.20 \hminfamily \showthe\font
```

4. get the family (hmin) and use it! (next slide)

Fonts usage

global

```
\usepackage{humanist}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\renewcommand{\rmdefault}{hmin}
{\LARGE fi} % ligature
the quick brown fox jumps over the lazy
↪  dog
\bfseries the quick brown fox jumps
↪  over the lazy dog
```

fi

the quick brown fox jumps over
the lazy dog

the quick brown fox jumps
over the lazy dog

THE QUICK BROWN FOX JUMPS OVER
THE LAZY DOG

0123456789

`\renewcommand{\rmdefault}<family_name>`

Local font usage throw package with pdfLaTeX, when the package is constructed to change defaults

Algorithm

You need to figuraute the Font Family

1. Check the package documentation
2. If manual is unreachable, get the Family directly: `\rmdefault` or `\familydefault`

```
\usepackage{berenis}
\usepackage[LY1]{fontenc}
\usepackage[utf8]{inputenc}

\rmdefault\ or \familydefault
```

ybd2j or ybd2j

3. remember the family (ybd2j) to use it (next slide)

Algorithm

```
\usepackage{berenis}
\usepackage[LY1]{fontenc}
\usepackage[utf8]{inputenc}

\renewcommand{\encodingdefault}{OT1}
\renewcommand{\rmdefault}{cmr}

{\LARGE fi} (default)\\
\fontfamily{ybd2j}\fontencoding{LY1}\selectfont
↪  {\LARGE fi} % ligature
```

fi (default)
fi

the quick brown fox jumps over
the quick brown fox jumps over the lazy dog
the quick brown fox jumps
the quick brown fox jumps over the lazy dog
THE QUICK BROWN FOX JUMPS
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789 0123456789

4. Change the encoding and font family to defaults (`\renewcommand{\encodingdefault}{OT1}`, `\renewcommand{\rmdefault}{cmr}`)

/25249/how-do-i-use-a-particular-font-for-a-small-section-of-text-in-my-document/25251#25251
http://mirrors.ctan.org/macros/latex/doc/fntguide.pdf: 2.2

Fonts usage

locally

How to change font:

- `\fontencoding` will change the encoding
- `\fontfamily` will change family
- `\fontseries` wil change series
- `\fontshape` will change shape
- `\fontsize` will change font size

... and `\selectfont` after font change!

We need `\selectfont` because while changing the font we can be in an inconsistent state: for example, we change the encoding, but now there is no such family as an old one!

/25249/how-do-i-use-a-particular-font-for-a-small-section-of-text-in-my-document/25251#25251
http://mirrors.ctan.org/macros/latex/doc/fntguide.pdf: 2.2

XeLaTeX and LuaTeX

Fonts usage

XeLaTeX and LuaTeX

1. You can use practically all fonts from pdfLaTeX
2. You can use OpenType (OTF), TrueType (TTF) fonts. They usually install in your system.

Global font usage throw global-available font with XeLaTeX

Font usage

default

```
\usepackage{fontspec}
{\LARGE ``fi''} % ligature
the quick brown fox jumps over the lazy
↪ dog
\bfseries the quick brown fox jumps
↪ over the lazy dog
THE QUICK BROWN FOX JUMPS OVER THE LAZY
↪ DOG
0123456789
```

“fi”

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789

\usepackage{fontspec}

Font usage

global

```
\usepackage{fontspec}
\setmainfont[Ligatures=TeX]{Arial}
{\LARGE ``fi''} % ligature
the quick brown fox jumps over the lazy
↪  dog
\bfseries the quick brown fox jumps
↪  over the lazy dog
THE QUICK BROWN FOX JUMPS OVER THE LAZY
↪  DOG
0123456789
```

“fi”

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789

`\setmainfont`<font-name>

/Fonts#Using_TTF_and_OTF_fonts /XeLaTeX

/25249/how-do-i-use-a-particular-font-for-a-small-section-of-text-in-my-document/37251#37251

Font usage

global

```
\usepackage{fontspec}
\setmainfont[Ligatures=TeX]{Georgia}
{\LARGE ``fi''} % ligature
the quick brown fox jumps over the lazy
↪ dog
\bfseries the quick brown fox jumps
↪ over the lazy dog
THE QUICK BROWN FOX JUMPS OVER THE LAZY
↪ DOG
0123456789
```

“fi”

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789

`\setmainfont`<font-name>

/Fonts#Using_TTF_and_OTF_fonts /XeLaTeX

/25249/how-do-i-use-a-particular-font-for-a-small-section-of-text-in-my-document/37251#37251

Font usage

global

- `\setmainfont` sets the roman font
- `\setsansfont` sets the sans font
- `\setmonofont` sets the monospace font

/Fonts#Using_TTF_and_OTF_fonts /XeLaTeX

/25249/how-do-i-use-a-particular-font-for-a-small-section-of-text-in-my-document/37251#37251

Local font usage throw global-avaliable font with XeLaTeX

Font usage

local

```
\usepackage{fontspec}
{\LARGE ``fi''} (default)\\
\fontspec[Ligatures=TeX]{Arial}\selectfont
↪  {\LARGE ``fi''} % ligature

\bfseries the quick brown fox jumps\\
\fontspec[Ligatures=TeX]{Arial}\selectfont
↪  \bfseries the quick brown fox jumps
↪  over the lazy dog
```

“fl” (default)
“fi”

the quick brown fox jumps over
the quick brown fox jumps over the lazy dog
the quick brown fox jumps
the quick brown fox jumps over the lazy dog
THE QUICK BROWN FOX JUMPS
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
0123456789 0123456789

`\fontspec<font-name>`

Font usage

local

```
\usepackage{fontspec}
\newfontfamily\myfont[Ligatures=TeX]{Arial}
{\LARGE ``fi''} (default)\\
\myfont\selectfont {\LARGE ``fi''} %
↪  ligature
\bfseries the quick brown fox jumps\\
\myfont\selectfont \bfseries the quick
↪  brown fox jumps over the lazy dog
```

“fi” (default)
“fi”

the quick brown fox jumps over
the quick brown fox jumps over the lazy dog
the quick brown fox jumps
the quick brown fox jumps over the lazy dog
THE QUICK BROWN FOX JUMPS
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG
0123456789 0123456789

`\newfontfamily` — more effective way

Global font usage throw local-avaliable font with XeLaTeX

Font usage

global

```
\usepackage{fontspec}
\setmainfont[Path = fontDir/,
↪  Ligatures=TeX,
↪  BoldFont=Lato-Bold.ttf]{Lato-Regular.ttf}

{\LARGE ``fi''} % ligature

the quick brown fox jumps over the lazy
↪  dog

\bfseries the quick brown fox jumps
↪  over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY
↪  DOG

0123456789
```

\setmainfont<font-filename>

“fi”

the quick brown fox jumps over the lazy dog

the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789

/XeLaTeX
from https://www.fontsquirrel.com/fonts/lato

fontspec's commands optional params

- BoldFont = font name
- ItalicFont = font name
- BoldItalicFont = font name
- SlantedFont = font name
- BoldSlantedFont = font name
- SmallCapsFont = font name
- UprightFont = font name

Local font usage throw local-avaliable font with XeLaTeX

Font usage

local

```
\usepackage{fontspec}
{\LARGE ``fi''} (default)\\
\fontspec[Path = fontDir/,
↪  Ligatures=TeX,
↪  BoldFont=Lato-Bold.ttf]{Lato-Regular.ttf}\selectfont
↪  {\LARGE ``fi''} % ligature
\bfseries the quick brown fox jumps\\
\fontspec[Path = fontDir/,
↪  Ligatures=TeX,
↪  BoldFont=Lato-Bold.ttf]{Lato-Regular.ttf}\selectfont
↪  \bfseries the quick brown fox jumps
↪  over the lazy dog
```

“ﬁ” (default)

“ﬁ”

the quick brown fox jumps over
the quick brown fox jumps over the lazy dog

the quick brown fox jumps
the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS
THE QUICK BROWN FOX JUMPS OVER
THE LAZY DOG

0123456789 0123456789

`\fontspec`<font-name>

Font usage

local

```
\usepackage{fontspec}
\newfontfamily\myfont[Path = fontDir/,
↪  Ligatures=TeX,
↪  BoldFont=Lato-Bold.ttf]{Lato-Regular.ttf}
{\LARGE ``fi''} (default)\\
\myfont\selectfont {\LARGE ``fi''} %
↪  ligature
\bfseries the quick brown fox jumps\\
\myfont\selectfont \bfseries the quick
↪  brown fox jumps over the lazy dog
```

“fi” (default)

“fi”

the quick brown fox jumps over
the quick brown fox jumps over the lazy dog

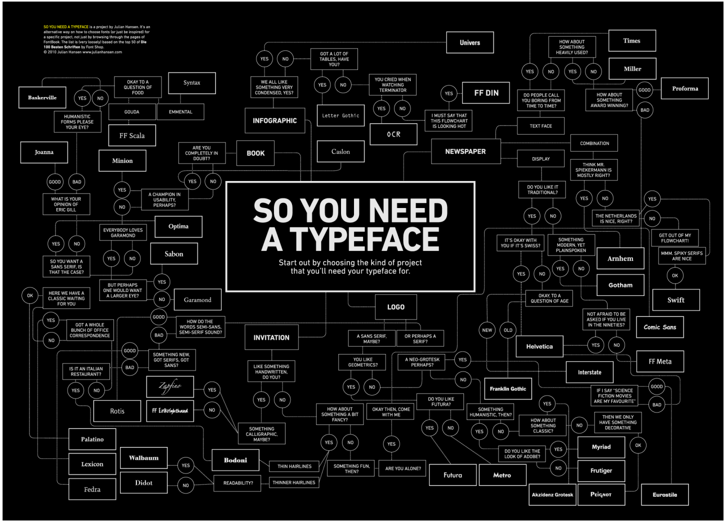
the quick brown fox jumps
the quick brown fox jumps over the lazy dog

THE QUICK BROWN FOX JUMPS
THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG

0123456789 0123456789

`\newfontfamily` — more effective way

How to find a font



https://www.labnol.org/home/choose-fonts-with-flowchart/13488/

Skoltech

Useful links

- https://tug.org/FontCatalogue/allfonts.html avaliable at LaTeX fonts
- https://www.fontsquirrel.com/ font catalogue
- https://www.fontsquirrel.com/matcherator identify font by picture
- https://www.fonts-online.ru/fonts/russian fonts with cyrillic
- http://allfont.ru/free/ fonts with cyrillic
- https://fonts.google.com/?subset=cyrillic fonts with cyrillic
- https://wordmark.it/ quick look of how your text will look like

Useful tips: Font pairs

Don't use too many fonts in your document! The best choice is two-three different fonts.

How to choose font pairs?

- https://www.fastprint.co.uk/blog/the-art-of-mixing-typefaces.html cheat list
- https://www.canva.com/font-combinations/ combinator
- https://fontpair.co/ list
- http://font-combinator.com/ combinator
- http://www.joustmultimedia.com/blog/post/the-art-of-combining-fonts some tips

from https://freelance.today/poleznoe/15-servisov-dlya-podbora-shriftovyh-par.html

What we have learned today?

Technical agreements

TikZ
- Introduction: what is TikZ and when to use it
- General usage
- Graphs
- Arrangment

(User-level) Typography
- Text position
- Other
- Fonts

references I

color from the footnotes corresponds to references' color.

- kn: Knuth "The TEXBook"
- lv: L'vovsky "Nabor i verstka v sisteme LATEX"
- lamport: Lamport. "LATEX. A Document Preparation System, User's Guide and Reference Manual"
- man: "LATEX2e: An unofficial reference manual" also at website https://latexref.xyz/
- : https://tex.stackexchange.com/questions
- : https://en.wikibooks.org/wiki/LaTeX
- : https://www.overleaf.com/learn/latex
- : https://www.tug.org/utilities/plain/cseq.html

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