LaTeX: from dummy to TeXnician

Typography

Anton Lioznov

Skoltech

ISP 2019,
lesson 4

What we will know?

Introduction

Package-based typography

Core-TEX typography

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

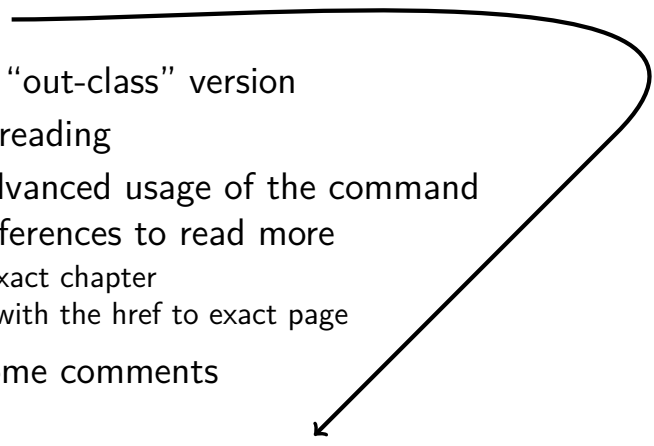
Anna Pavlicheva for counseling of non-techniqual part about fonts

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

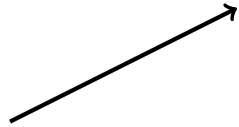


Like this

Agreements

II

Addition information – “magic”



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

What we have learned today?

What we will know?

Introduction

Package-based typography

Core-TEX typography

Why we are studying it?

What with typography?

For most of “simple” articles you need no typography knowledge: TEX will do all by itself.

But for something complex, like CV, presentations,... You need more!.

And sometimes you really need to “break the rules”

What we will know?

Introduction

Package-based typography

Core-TEX typography

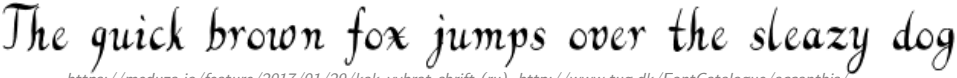
What we will know?

Package-based typography

- Fonts
- Text position
- Other

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



https://meduza.io/feature/2017/01/29/kak-vybrat-shrift (ru), 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

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

then just follow the instructions for the package

Fonts usage

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
```

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{fontenc}

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
```

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

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
```

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

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

Fonts usage

Understanding warning

Last example provides warning:

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

Sometimes you can find 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

Understanding warning

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

Understanding warning

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

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

1.20 \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>
```


/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: 2.4

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}{OT
    1}
\renewcommand{\rmdefault}{cmr}
{\LARGE fi} (default)\\
\fontfamily{ybd2j}\fontencoding{LY
    1}\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>

Font usage

global

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

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
```

“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 usag

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
```

“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-filename>

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

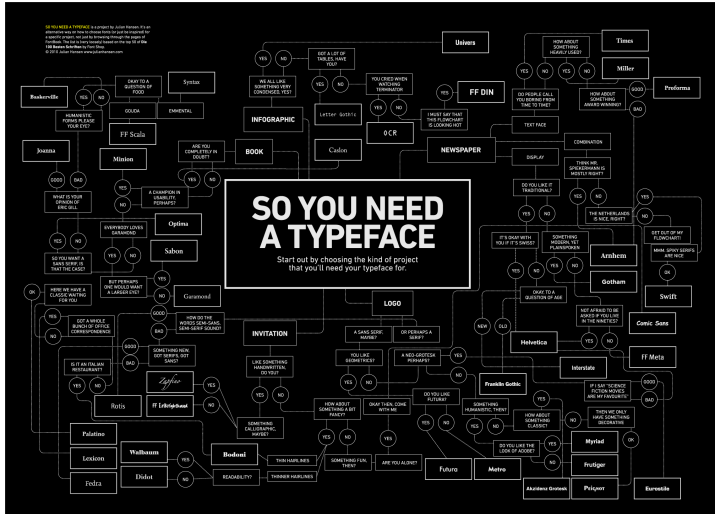
the quick brown fox jumps
the quick brown fox jumps over the lazy dog

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/

Useful links

- http://www.tug.dk/FontCatalogue/ 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 will know?

Package-based typography

- Fonts
- Text position
- Other

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

Yell
o Hell
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}
```

Hello!

world!

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

What we will know?

Package-based typography
- Fonts
- Text position
- Other

Geometry

```
\usepackage{geometry}
\geometry{a4paper,paperwidth=70mm, paperheight=65mm,
left=5mm, top=5mm, 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?

Introduction

Package-based typography

Core-TEX typography

What we will know?

Core-TEX typography

- Lengths
- Boxes
- Glue
- Struts etc
- Modes
- Paragraphs and pages creation
- Other

Length

absolute values

		most common used:	
pt	points	≃0.35mm	 12pt
mm	millimeters	≃2.84pt	 10mm
cm	centimeter	≃28.4pt, 10mm	 1cm
in	inch	≃72.27pt, ≃, 25.4mm	 1in

Length

absolute values

		not so common used:	
pt	points	≃0.35mm	12pt
mm	millimeters	≃2.84pt	10mm
bp	big point	1/72in, ≃1.003pt	12bp
pc	pica	12pt, 4.2mm	1pc
dd	didot	≃1.07pt, ≃0376mm	12dd
cc	cicero	12dd	1cc
sp	scaled point	$1/2^{16}$pt = 1/65536pt	2097152sp

(pt and mm here for comparison)

Every TEX's length is a integer number of sp

Length

Relative values

pt	points	12pt
mm	millimeters	10mm
em	roughly the width of an 'M' (uppercase)	1em
ex	roughly the height of an 'x'	1ex

example: \Huge

mm	10mm
em	1em

use em for horizontal and ex for vertical cases

Prebuild lengths

Most used

TEX's	
\parindent	The normal paragraph indentation
\parskip	The extra vertical space between paragraphs

LATEX's	
\textwidth	The width of the text on the page
\textheight	The height of the text on the page
\linewidth	The width of the text in this “box”
\lineheight	The height of the text in this “box”

By typography rules, don't put both parskip and parindent as paragraph separation.

Lengths#LaTeX_default_lengths man: 5.5 /16942/difference-between-textwidth-linewidth-and-hsize

\parskip is actually a glue. You can define your own lengths (See next lecture.)

Prebuild lengths

not so common used

LaTeX

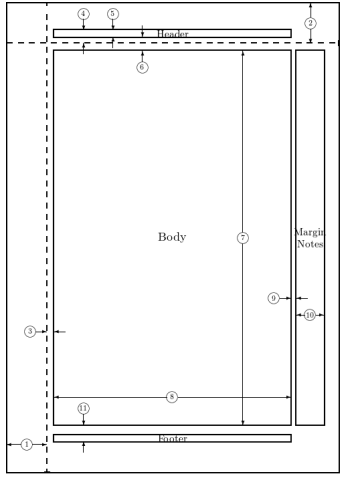
practicully every length — the margins; footnote, footer/header place; distance between columns,..

TeX

`\hsize`, `\vsize`	the normal size of text in page
`\hoffset`, `\voffset`	the offset of on page

by the way, you can magnitude whole text by `\mag=...` *parameter. Also there are* `\hangindent` *and* `\hangafter` *TeX params, but we'll speak in paragraph section.*

lv: IV.4 man: 5.5




```
1   one inch + \hoffset          2   one inch + \voffset
3   \oddsidemargin = 13pt        4   \topmargin = -23pt
5   \headheight = 12pt           6   \headsep = 25pt
7   \textheight = 674pt          8   \textwidth = 426pt
9   \marginparsep = 10pt         10  \marginparwidth = 50pt
11  \footskip = 30pt                 \marginparpush = 5pt (not shown)
    \hoffset = 0pt                   \voffset = 0pt
    \paperwidth = 597pt              \paperheight = 845pt
```

How to use lengths

```
\usepackage{printlen}
\indent\printlength{\parindent}\
    par
\parindent=1pt\indent\printlength
    {\parindent}\par
\parindent=4em\indent\printlength
    {\parindent}\par
```

15.0pt
1.0pt
40.0pt

Just `\<length-command>=<length>`

How to use lengths

```
\usepackage{printlen}
\indent\printlength{\parindent}\
    par
\parindent=1pt\indent\printlength
    {\parindent}\par
\parindent=4em\indent\printlength
    {\parindent}\par
\parindent=0.5\parindent\indent\
    printlength{\parindent}\par
```

15.0pt
1.0pt
40.0pt
20.0pt

Arifmetics: <multiply-factor>\<length-command>

How to use lengths

```
\usepackage{printlen}
\indent\printlength{\parindent}\
    par
\parindent=1pt\indent\printlength
    {\parindent}\par
\parindent=4em\indent\printlength
    {\parindent}\par
\parindent=0.5\parindent\indent\
    printlength{\parindent}\par
\parindent=2em+1cm\indent\
    printlength{\parindent}\par
```

15.0pt
1.0pt
40.0pt
20.0pt
+1cm 20.0pt

Arifmetics: BUT You can't use simple notation +, -, /, *,...

How to use lengths

```
\usepackage{printlen}
\indent\printlength{\parindent}\
    par
\parindent=1pt\indent\printlength
    {\parindent}\par
\parindent=4em\indent\printlength
    {\parindent}\par
\parindent=0.5\parindent\indent\
    printlength{\parindent}\par
\parindent=2em+1cm\indent\
    printlength{\parindent}\par
\parindent=\dimexpr2em+1cm\indent\
    printlength{\parindent}\par
\parindent=\dimexpr(2em+1cm)/2\
    indent\printlength{\parindent
    }\par
```

15.0pt
1.0pt
40.0pt
20.0pt
+1cm 20.0pt
48.45274pt
24.22638pt

Arifmetics: `\dimexpr` allow to use “normal” notation.

What we will know?

Core-TEX typography

- Lengths
- Boxes
- Glue
- Struts etc
- Modes
- Paragraphs and pages creation
- Other

Main idea

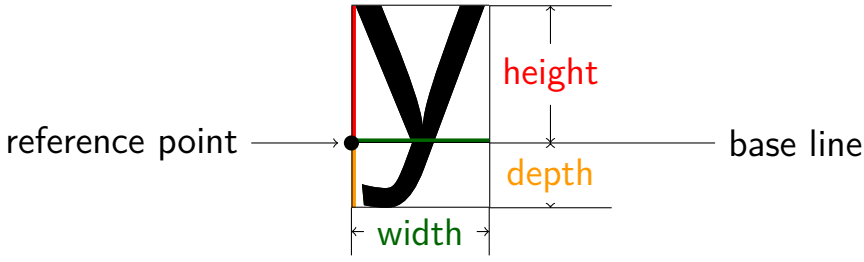
Boxes and Glue

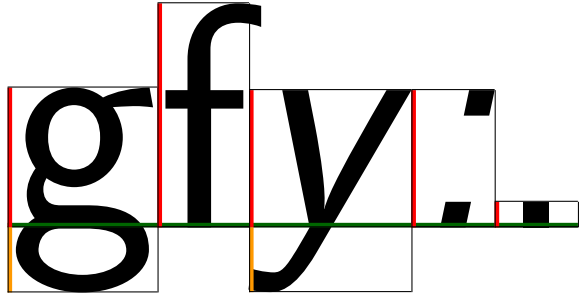
Main idea

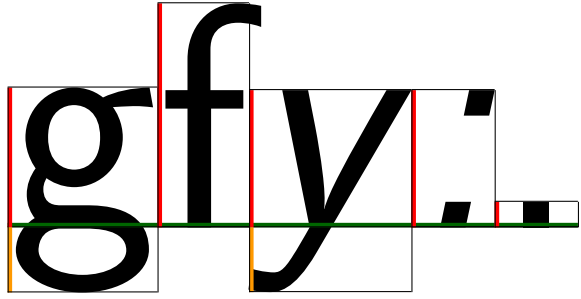
A symbol is a box
it is part of a word, that is a box
the words are connected with glue
into sentances and paragraphs.
A paragraph is a box
it connected with another one with glue
to the page. Which is a box

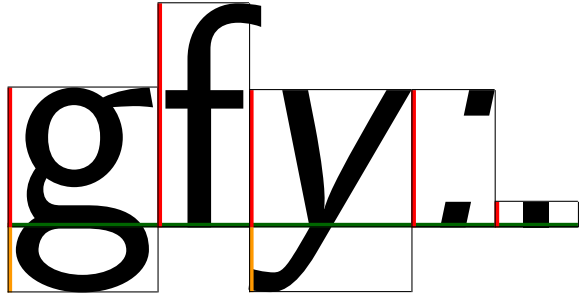
by the way: table, picture, ... is a box

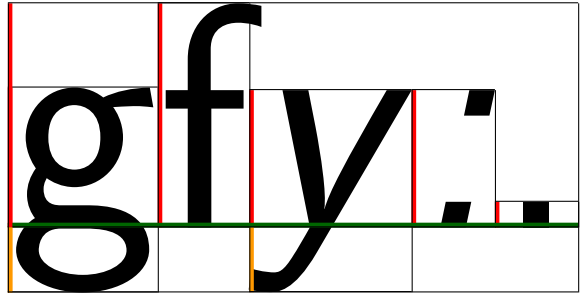
Box params









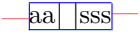


TEX boxes

“Horizontal” boxes

TEX-way

```
\boxing{\hbox          {aa| |sss}}\\
```

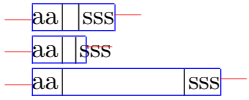


\hbox create box around the text. The box will never split in linebreak.

“Horizontal” boxes

TEX-way

```
\boxing{\hbox         {aa| |sss}}\\
\boxing{\hbox to 20pt{aa| |sss}}\\
\boxing{\hbox to 70pt{aa| |sss}}\\
```

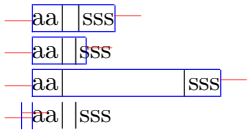


You can specify the length of the box with keyword `to`

“Horizontal” boxes

TEX-way

```
\boxing{\hbox          {aa| |sss}}\\
\boxing{\hbox to 20pt{aa| |sss}}\\
\boxing{\hbox to 70pt{aa| |sss}}\\
\boxing{\hbox to -4pt{aa| |sss}}\\
```

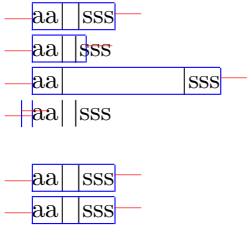


You even can set the box to negative size

“Horizontal” boxes

TEX-way

```
\boxing{\hbox          {aa| |sss}}\\
\boxing{\hbox to 20pt{aa| |sss}}\\
\boxing{\hbox to 70pt{aa| |sss}}\\
\boxing{\hbox to -4pt{aa| |sss}}\\
\boxing{\hbox                {aa| |
    sss}}\\
\boxing{\hbox spread 0pt  {aa| |
    sss}}\\
```

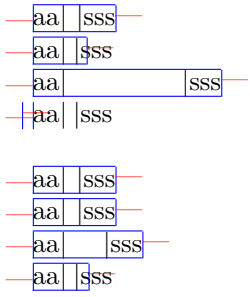


Another keyword, spread is the addition width

“Horizontal” boxes

TEX-way

```
\boxing{\hbox          {aa| |sss}}\\
\boxing{\hbox to 20pt{aa| |sss}}\\
\boxing{\hbox to 70pt{aa| |sss}}\\
\boxing{\hbox to -4pt{aa| |sss}}\\
\boxing{\hbox                {aa| |
    sss}}\\
\boxing{\hbox spread 0pt  {aa| |
    sss}}\\
\boxing{\hbox spread 10pt {aa| |
    sss}}\\
\boxing{\hbox spread -10pt {aa| |
    sss}}\\
```



also both positive and negative

“Horizontal” boxes

Usage

```
\hbox to -1pt{/}=
```

$\neq$

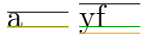
```
\begin{tabbing}
\hbox to 4em{}\=\hbox to 4em{}\kill
a \> b\\
hello\> world!
\end{tabbing}
```

a	b
hello	world!

“Vertical” boxes

TEX-way

```
\boxing{\vbox{a}}
\boxing{\vbox{yf}}
```



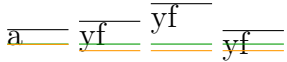
\vbox create box around the text

#vbox-rp, #vtop-rp

“Vertical” boxes

TEX-way

```
\boxing{\vbox{a}}
\boxing{\vbox{yf}}
\boxing{\vbox to 15pt{yf}}
\boxing{\vbox to 5pt{yf}}
```



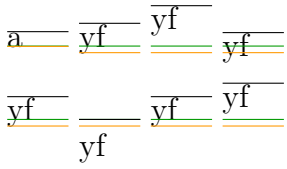
You can specify the height of the box with keyword `to`

#vbox-rp, #vtop-rp

“Vertical” boxes

TEX-way

```
\boxing{\vbox{a}}
\boxing{\vbox{yf}}
\boxing{\vbox to 15pt{yf}}
\boxing{\vbox to 5pt{yf}}
\boxing{\vbox{yf}}
\boxing{\vbox to -5pt{yf}}
\boxing{\vbox spread 0pt{yf}}
\boxing{\vbox spread 5pt{yf}}
```



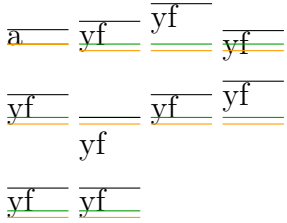
You even can set the box to negative size, use `spread`. But depth will remain the same

#vbox-rp, #vtop-rp

“Vertical” boxes

TEX-way

```
\boxing{\vbox{a}}
\boxing{\vbox{yf}}
\boxing{\vbox to 15pt{yf}}
\boxing{\vbox to 5pt{yf}}
\boxing{\vbox{yf}}
\boxing{\vbox to -5pt{yf}}
\boxing{\vbox spread 0pt{yf}}
\boxing{\vbox spread 5pt{yf}}
\boxing{\vbox{yf}}
\boxing{\vtop{yf}}
```



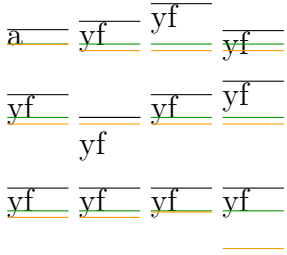
There is another box, \vtop

#vbox-rp, #vtop-rp

“Vertical” boxes

TEX-way

```
\boxing{\vbox{a}}
\boxing{\vbox{yf}}
\boxing{\vbox to 15pt{yf}}
\boxing{\vbox to 5pt{yf}}
\boxing{\vbox{yf}}
\boxing{\vbox to -5pt{yf}}
\boxing{\vbox spread 0pt{yf}}
\boxing{\vbox spread 5pt{yf}}
\boxing{\vbox{yf}}
\boxing{\vtop{yf}}
\boxing{\vtop spread -2pt{yf}}
\boxing{\vtop to 20pt{yf}}
```



it will change the depth for you

#vbox-rp, #vtop-rp

Move boxes

```
Write as L\raise0.5ex\hbox{A}T\
    lower0.5ex\hbox{E}X
```

Write as $\mathrm{L^{A}T_{E}X}$

`\raise` and `\lower` for horizontal boxes

Move boxes

```
Write as L\raise0.5ex\hbox{A}T\
    lower0.5ex\hbox{E}X
\vspace{3ex}
\vbox{L}
\moveleft0.5em\vbox{A}
\vbox{T}
\moveright0.5em\vbox{E}
\vbox{X}
```

Write as LATEX

L
A
T
E
X

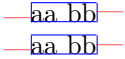
\moveleft and \moveright for vertical boxes

#raise-rp #lower-rp #moveleft-rp #moveright-rp

LaTeX boxes

Horizontal boxes

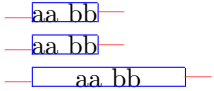
```
\boxing{\mbox{aa bb}}
\boxing{\makebox{aa bb}}
```



\mbox and \makebox are like \hbox

Horizontal boxes

```
\boxing{\mbox{aa bb}}
\boxing{\makebox{aa bb}}
\boxing{\makebox[20mm]{aa bb}}
```

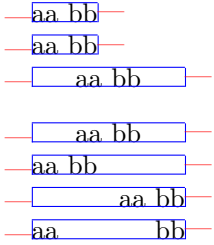


`\makebox` has a width as an optional param

man: 20.1 Boxes#makebox_and_mbox

Horizontal boxes

```
\boxing{\mbox{aa bb}}
\boxing{\makebox{aa bb}}
\boxing{\makebox[20mm]{aa bb}}
\boxing{\makebox[20mm][c]{aa bb}}
\boxing{\makebox[20mm][l]{aa bb}}
\boxing{\makebox[20mm][r]{aa bb}}
\boxing{\makebox[20mm][s]{aa bb}}
```



... and \makebox has text location as second optional param

man: 20.1 Boxes#makebox_and_mbox

Paragraph boxes

```
Space space \parbox{8em}{wanna go
    to space yes please space.
    Space space.} Go to space.
```

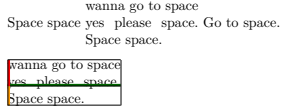
wanna go to space
Space space yes please space. Go to space.
Space space.

\parbox give you a box of text with some width. Also there are \pbox and minipage envirument

man: 20.3 man: 8.18*73* Boxes#parbox,_minipage,_and_pbox

Paragraph boxes

```
Space space \parbox{8em}{wanna go
    to space yes please space.
    Space space.} Go to space.
\boxingDim{\parbox{8em}{wanna go
    to space yes please space.
    Space space.}}
```

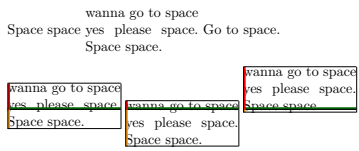


It is just a box with a big depth

man: 20.3 man: 8.18*73* Boxes#parbox,_minipage,_and_pbox

Paragraph boxes

```
Space space \parbox{8em}{wanna go
    to space yes please space.
    Space space.} Go to space.
\boxingDim{\parbox{8em}{wanna go
    to space yes please space.
    Space space.}}
\boxingDim{\parbox[t]{8em}{wanna
    go to space yes please space.
    Space space.}}
\boxingDim{\parbox[b]{8em}{wanna
    go to space yes please space.
    Space space.}}
```



You can specify the position.

`\parbox` is useful when you want to put two lines to some command, that accepts only one line. Footnotes in the lectures use it.

man: 20.3 man: 8.18*73* Boxes#parbox,_minipage,_and_pbox

Boxes-modifiers

```
\raisebox{0pt}[0pt][0pt]{\Large%
  \textbf{Aaaa\raisebox{-0.3ex}{a}%
    \raisebox{-0.7ex}{aa}%
    \raisebox{-1.2ex}{r}%
    \raisebox{-2.2ex}{g}%
    \raisebox{-4.5ex}{h}
  }
}
he shouted.
\rotatebox{45}{A}
\scalebox{2}{A}
```

Aaaaaaaarg
h

he shouted.

A

\raisebox{lift}[height][depth]{text} change the text position. \rotatebox rotates the text, \scalebox scales it

man: 20.4 man: 22.3.2 man: 22.3.3 Boxes#box_modifiers

What we will know?

Core-TEX typography

- Lengths
- Boxes
- Glue
- Struts etc
- Modes
- Paragraphs and pages creation
- Other

Spaces

`glue` and `kern` provides spaces between boxes.

```
G\hskip0em lu\hskip0.5em e and k\kern
    0em e\kern0.5em rn provides...
```

Glu e and ke rn provides...

What is glue

Glue is more than just “spaces” between the boxes.
Glue is a **tensile** spaces between boxes.
Glue syntax is: <normal-length> [plus <how-can-it-stretch>] [minus <how-can-it-shrink>].

\relax

Don't want unexpected adding to you glue? (Or two you commands?) Use `\relax`! `\relax` does nothing by itself but says TEX "This is the end of what you've been doing"

Where glue adds implicitly?

Between words and sentences. Here are lots' of glue.
Between words and sentences. Here are lots' of glue.
Between words and sentences. Here are lots' of glue.
Between words and sentences. Here are lots' of glue.
Between words and sentences. Here are lots' of glue.
Between words and sentences. Here are lots' of glue.
Between words and sentences. Here are lots' of glue.
Between words and sentences. Here are lots' of glue.
Between words and sentences. Here are lots' of glue.

P.S. here is \hbox spread in range -40pt—40pt Between paragraphs, there is also a glue. Notice: between sentences the glue is bigger, than between words.

How to use glue in your own work

```
\TeX-way:
use \hskip2em plus 1em\relax to add
    horizontal glue

or in vertical mode
\vskip2em plus 1em\relax
\LaTeX-way:
use \hspace{2em plus 1em} to add
    horizontal glue

or in vertical mode
\vspace{2em plus 1em}

like this
```

TEX-way: use to add horizontal glue
or in vertical mode

LATEX-way: use to add horizontal glue
or in vertical mode

like this

For horizontal space use `\hskip` or `\hspace`
For vertical space use `\vskip` or `\vspace`.

#hskip-rp #vskip-rp man: 19.2 man: 19.14
use \vspace∗ if you want space at begin of the page!

Infinite glue

```
\hbox to 50mm{\hskip0em plus 1fil\
    relax 1fil and 1fil \hskip0em
    plus 1fil\relax}
\hbox to 50mm{\hskip0em plus 1fil\
    relax 1fil and 2fil \hskip0em
    plus 2fil\relax}
\hbox to 50mm{\hskip0em plus 1fill\
    relax fill and fill \hskip0em
    plus 1fill\relax}
\hbox to 50mm{\hskip0em plus 1fill\
    relax fill vs fil \hskip0em plus
    999fil\relax}
\hbox to 50mm{\hskip0em plus 1filll\
    relax filll and filll \hskip0em
    plus 1filll\relax}
\hbox to 50mm{\hskip0em plus 1filll\
    relax filll vs fill \hskip0em
    plus 999fill\relax}
```

1fil and 1fil
1fil and 2fil
fill and fill
fill vs fil
filll and filll
filll vs fill

`fil`, `fill`, `filll` are infinity with different “power”. Both “plus” and “minus” are allowed. Notice: alignment without tabbular!

Abbreviations

You can use:

`\hfil`	`\hfill`	`\hspace{\fil}`	`\hspace{\fill}`
`\vfil`	`\vfill`	`\vspace{\fil}`	`\vspace{\fill}`

What we will know?

Core-TEX typography

- Lengths
- Boxes
- Glue
- Struts etc
- Modes
- Paragraphs and pages creation
- Other

What spaces we have?

\qquad

\quad

\enspace

\

\quad is used below:

$x = y \quad$ if y=0

useful for math:

\;

\>

\,

\!

the last one is negative space

Phantoms

Phantoms have the same size as it's an argument without drawing.

```
\boxing{$\int\limits^a_b x d\!x$}
\boxing{\phantom{$\int\limits^a_b x d
    \!x$}}
\boxing{\hphantom{$\int\limits^a_b x
    d\!x$}}
\boxing{\vphantom{$\int\limits^a_b x
    d\!x$}}
\boxing{\strut}
\hfill
\boxing{\smash{$\int\limits^a_b x d\!
    x$}}
```

`\phantom` leaves both dimentions. `\hphantom` and `\vphantom` leaves only one dimention.

`\strut` is short for `\vphantom{()}`

`\smash` is using to leave only the horizontal coordinate of a formula

What we will know?

Core-TEX typography

- Lengths
- Boxes
- Glue
- Struts etc
- Modes
- Paragraphs and pages creation
- Other

Modes

TEX has 3(6) modes:

1. **Vertical mode.** [Building the main vertical list, from which the pages of output are derived.]
2. **Internal vertical mode.** [Building a vertical list for a vbox.]
3. **Horizontal mode.** [Building a horizontal list for a paragraph.]
4. **Restricted horizontal mode.** [Building a horizontal list for an hbox.]
5. **Math mode.** [Building a mathematical formula to be placed in a horizontal list.]
6. **Display math mode.** [Building a mathematical formula to be placed on a line by itself, temporarily interrupting the current paragraph.]

Difference between modes

The modes have lots of differences. For example:

- in horizontal mode only first space is taking into account
- in math mode generic font is italic, all spaces are ignored
- in Display math mode operators are drawing bigger, than in the regular one
- in vertical mode all spaces and <return>s are ignored

use `\leavevmode` *to leave vertical mode*

More about math mode

Math actually has 4 different styles. When you see that superscript x^y is smaller then the text — it is a different style. The styles are:

Display style	`\displaystyle`	A	main style for displayed formula
Text style	`\textstyle`	A	main style for in-text formula
Script style	`\scriptstyle`	A	main style for scripts
Script-script style	`\scriptscriptstyle`	A	main style for scripts in scripts

What we will know?

Core-TEX typography

- Lengths
- Boxes
- Glue
- Struts etc
- Modes
- Paragraphs and pages creation
- Other

Slide for perfectionists

how many word-breaks are

Microsoft Word 2008

Call me Ishmael. Some years ago – never mind how long precisely – having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen, and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin

Adobe InDesign CS4

Call me Ishmael. Some years ago – never mind how long precisely – having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen, and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin warehous-

pdf-LaTeX 3.1415926

Call me Ishmael. Some years ago – never mind how long precisely – having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of driving off the spleen, and regulating the circulation. Whenever I find myself growing grim about the mouth; whenever it is a damp, drizzly November in my soul; whenever I find myself involuntarily pausing before coffin

Hyphenation and inter-word spacing statistics

	Word	InDesign	pdf-LaTeX
Number of hyphenations	9	10	4
SD of IWS (pt)	2.26	1.94	1.42
Maximum IWS (pt)	14.4	13.2	9.0
Number of lines with IWS > 9 pt	5	2	0

SD: standard deviation; IWS: inter-word spacing

/110133/visual-comparison-between-latex-and-word-output-hyphenation-typesetting-ligat

http://www.rtznet.nl/zink/latex.php?lang=en

“this, in fact, is probably the most interesting aspect of the whole TEX system”

D. Knuth, the TEXBook

Paragraph creation overview

- All paragraph is considered as one: the words in the last line can change the typesetting in the first line.
- TEX will never put words narrow than the glue allow.
- TEX tries out all possible varients for line breaks. For each varient and each line TEX calculates the *badness*. If it is lower than `\tolerance`, TEX will try to create paragraph with the minimum of hyphenation.
- if TEX fails, it provides **Overfull** or **Underfull** warnings.

How to suggest a hyphentation

Locally: use `\-` as in this ve\-ry long se\-nta\-nce
Globally: `\hyphenation{some-thing poss-ible}`

`\-` is just a short version for `\discretionary{hpre-break texti}{hpost-break texti}{hno-break texti}
also: TEX will never hyphenate the word with "/". Use `\slash`if you want to allow it. And `\uchyph=0` will prohibit hyphenation in words on uppercase letter.

Manual line break manipulation

Never break: non-breaking space `~`, `\nobreak`, `\nolinebreak`
Always: `\\`, `\break`, `\linebreak`
You can use `\obeylines` to follow the line breaks in the source code.

by the way: `\\` has an optional parameter: the vertical space after the command. Also `\smallskipamount`, `\medskipamount` and `\bigskipamount` responsible for skipping after paragraph. `\linebreak` has an optional parameter: [0-4] how much you want to have the break here.

Algorithm: part 1

1. TEX produce varients without word breaks. It compare the *badness* with `\pretolerance` param.
2. *badness* is $\simeq 100 \cdot <$proportion-between-the-normal-glue-and-its-stretching/compression$>^3$
3. if `\pretolerance`-try fall, TEXwill try to use all posible line breaks to make each badness less than `\tolerance`

defaults: `\pretolerance`=100, `\tolerance`=200. `\pretolerance`>=10000, the first try will always succeed

Algorithm: part 2

1. line breaks are allowed only in certain places:
   - 1.1 glue
   - 1.2 kern with glue after
   - 1.3 and of math ($) and glue after
   - 1.4 the manual or auto-passed penalty
   - 1.5 discretionary break
2. The penalty to the first three is 0. For the last one, it is defined by `\hyphenpenalty=` or `\exhyphenpenalty=`. The penalty can be manually added as `\penalty`
3. Penalty can both positive and negative. If it is $> 10^4$ there will be no break ever, if it is $< -10^4$ there always will be a break

defaults: `\hyphenpenalty=50`, `\exhyphenpenalty=50`
you can put addition spaces between math and text with `\mathsurround`

Algorithm: part 3

1. in reality, TEX tries to minimize the *demerits*. It is proportional to the badnesses, `\linepenalty`(determines how much you want tex to produce a minimum amount of lines) and penalty
2. TEX also takes into account and add penalty if two lines one after another has a hyphenation (`\doublehyphendemerits`), if lines are *visually incompatible* (ex: if a tight line is next to a loose one) (`\adjdemerits`) and if the second-last line of the entire paragraph ends with a discretionary (`\finalhyphendemerits`)

defaults: `\linepenalty`*=10,* `\adjdemerits`*=10000,* `\doublehyphendemerits`*=10000,* `\finalhyphendemerits`*=5000.*
`\hfuzz`*=... will add the maximum length of string's aligment*

What else?

- Use `\narrow` to make lines narrow
- Use `\looseness=-1` to ask TEX to try make one line less in paragraph
- `\prevgraf` shows the curent line in the paragraph.
- `\vadjust` adds something at the vertical list after current line. With it we add the star to the left

*

- `\everypar`adds something in each paragraph
- `\parfillskip`— the glue after last line
- `\parskip`— the vertical glue between paragraphs

Non-standart paragraph form

```
\hangindent=1.5cm
\hangafter=-2 \noindent
With such paragraphs we can add
    something to the begin of the
    paragraph! It is really
    interesting.

\vspace*{\fill}

\hangindent=-1.5cm
\hangafter=1 \noindent
With such paragraphs we can add
    something to the end of the
    paragraph! It is really
    interesting.
```

With such paragraphs
we can add something
to the begin of the paragraph! It
is really interesting.

With such paragraphs we can add
something to the end
of the paragraph! It is
really interesting.

\hangindent=[>0] — the addition indent to the left. ...=[<0] — indent to the right.
\hangafter=[>0] — the indent to all lines after. ...=[<0] — before.

Non-standart paragraph form

```
\parshape=14
0cm 6cm .1cm 5.8cm .17cm 5.66cm .5cm
    5cm
.9cm 4.2cm 1.05cm 3.9cm 1.1cm 3.8cm
    1.1cm 3.8cm
1.05cm 3.9cm .9cm 4.2cm .5cm 5cm .17
    cm 5.66cm
.1cm 5.8cm 0cm 6cm
\noindent \small
Lorem ipsum dolor sit amet,
    consectetur adipiscing elit. Ut
    elit tellus, pharetra quis est ac
    , aliquam lobortis odio...
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, pharetra quis est ac, aliquam lobortis odio... Suspendisse at purus eu elit sagittis euismod sit amet id nulla. Quisque faucibus bibendum nisl ac commodo. Aliquam elit nisl, accumsan sit amet fermentum a, porttitor sit amet turpis. Sed a blandit leo, a suscipit nibh. Pellentesque non purus aliquam, rhoncus felis sed, accumsan nisi. Cras sed eros dapibus, blandit enim in, tempus massa. In tristique orci dui, eu porttitor mauris condimentum vitae.

Page creation

- TEX was created in time when it was not enough memory to optimize pages globally.
- TEX finds the best break to the current page and then erase it from memory.
- More or less the algorithms are the same.
- You can use `\penalty` or `\nobreak` in vertical mode
- You can use to remove bottom page alignment
- Also as in paragraph, you allow to use `\newpage`, `\pagebreak`, `\nopagebreak`

What we will know?

Core-TEX typography

- Lengths
- Boxes
- Glue
- Struts etc
- Modes
- Paragraphs and pages creation
- Other

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`

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

references II

-  : 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