



SKILLPILLS

Skill Pill: L^AT_EX Course

Lecture 3: Advanced Typesetting

James Schloss (cough Mark Kremer cough)

Okinawa Institute of Science and Technology
James.schloss@oist.jp

January 25, 2016



- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

Absolute Units:

cm

mm

pt 1 pt = 0.351459804 mm

in 1 inch = 2.54 cm

Relative Units:

em width of the capital 'M' in the font in use

ex heights of the lowercase 'x' in the font in use

Horizontal spacing:

`\hspace{length}` creates a horizontal space of the defined length except at the beginning or end of a line

`\hspace*{length}` similar to `\hspace{length}`, but it also works at the beginning or end of a line

`\hfill` creates a 'rubber length', which means that it will expand as much as needed to fill the line maximally.

Horizontal spacing:

`\hspace{length}` creates a horizontal space of the defined length except at the beginning or end of a line

`\hspace*{length}` similar to `\hspace{length}`, but it also works at the beginning or end of a line

`\hfill` creates a 'rubber length', which means that it will expand as much as needed to fill the line maximally.

`\dotfill` similar to `\hfill`, but with dots

`\hrulefill` similar to `\hfill`, but with a line

Vertical spacing:

`\vspace{length}` creates a vertical space of the defined length in between two paragraphs except at the beginning or end of a page

`\vspace*{length}` similar to `\vspace{length}`, but it also works at the beginning or end of a page

`\vfill` creates a 'rubber length', which means that it will expand as much as needed to fill the page maximally.

Vertical spacing:

`\vspace{length}` creates a vertical space of the defined length in between two paragraphs except at the beginning or end of a page

`\vspace*{length}` similar to `\vspace{length}`, but it also works at the beginning or end of a page

`\vfill` creates a 'rubber length', which means that it will expand as much as needed to fill the page maximally.

`\\[length]` creates a vertical space of the defined length after the line break

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

Counters are integers which are used for numbering various objects.

Important counters are:

`page` counts the pages

`section` counts the sections (c.f. subsection, subsubsection)

`equation` counts the numbered equations

`enumi` counts the numbered items of the `enumerate` environment
(first layer, `enumii`, `enumiii`, `enumiv`, for the deeper layers)

\LaTeX is organizing those counters automatically.

Counters can be manipulated using the following commands:

`\setcounter{counter}{value}` changes counter to value.

`\addtocounter{counter}{value}` adds value to counter.

`\stepcounter{counter}` increases the value of counter by 1 and resets all dependent counters to zero.

`\refstepcounter{counter}` similar to `\stepcounter{counter}`, but also adds a label for referencing.

In order to check the value of a set counter, you can call them in the document as follows

`\thepage` page count

`\thesection` section count (c.f. subsection, subsubsection)

`\theequation` equation count

`\thecount` current count in the latest enumerate environment

L^AT_EX will also call this command to print each counter, then increment the value by 1.

You can define your own counters using:

`\newcounter{myCounter}[higherrankcounter]` creates a new counter, with the name `myCounter` having the value 0. You can add an existing counter as optional argument (`higherrankcounter`) so that every time this counter is increased using `\stepcounter{higherrankcounter}` the new counter `myCounter` is set back to 0.

You can define your own counters using:

`\newcounter{myCounter}[higherrankcounter]` creates a new counter, with the name `myCounter` having the value 0. You can add an existing counter as optional argument (`higherrankcounter`) so that every time this counter is increased using `\stepcounter{higherrankcounter}` the new counter `myCounter` is set back to 0.

With `\value{counter}` you can use the value of the counter in the source code. This command is not for printing the value in the document, but rather for handling.

Example:

```
\setcounter{Counter2}{\value{Counter1}}
```

Counter2 is set to the value of Counter1

If you want to start the document from page 5, you can use

```
\setcounter{page}{5}
```

at the beginning of the document.

In order to print the counters you can use different commands:

`\arabic{counter}` prints the value of counter as an arabic number (4).

`\roman{counter}` prints the value of counter as an lower case roman number (iv).

`\Roman{counter}` prints the value of counter as an upper case roman number (IV).

`\alph{counter}` prints the value of counter as an lower case letter (d).

`\Alph{counter}` prints the value of counter as an upper case letter (D).

Similar to counters you can also create and manipulate lengths:

`\newlength{\name}` creates the length `\name` and sets it to 0pt.

Similar to counters you can also create and manipulate lengths:

`\newlength{\name}` creates the length `\name` and sets it to 0pt.

`\setlength{\name}{value}` sets the length of `\name` to value.

Similar to counters you can also create and manipulate lengths:

`\newlength{\name}` creates the length `\name` and sets it to 0pt.

`\setlength{\name}{value}` sets the length of `\name` to `value`.

`\addtolength{\name}{value}` adds `value` to `\name`.

Similar to counters you can also create and manipulate lengths:

`\newlength{\name}` creates the length `\name` and sets it to 0pt.

`\setlength{\name}{value}` sets the length of `\name` to `value`.

`\addtolength{\name}{value}` adds `value` to `\name`.

`\settowidth{\name}{text}` sets the value of `\name` to the width of `text` in the current font.

Similar to counters you can also create and manipulate lengths:

`\newlength{\name}` creates the length `\name` and sets it to 0pt.

`\setlength{\name}{value}` sets the length of `\name` to `value`.

`\addtolength{\name}{value}` adds `value` to `\name`.

`\settowidth{\name}{text}` sets the value of `\name` to the width of `text` in the current font.

`\settoheight{\name}{text}` sets the value of `\name` to the heights of `text` above the baseline.

Similar to counters you can also create and manipulate lengths:

`\newlength{\name}` creates the length `\name` and sets it to 0pt.

`\setlength{\name}{value}` sets the length of `\name` to `value`.

`\addtolength{\name}{value}` adds `value` to `\name`.

`\settowidth{\name}{text}` sets the value of `\name` to the width of `text` in the current font.

`\settoheight{\name}{text}` sets the value of `\name` to the heights of `text` above the baseline.

`\settodepth{\name}{text}` sets the value of `\name` to the heights of `text` below the baseline.

Lengths can for example be used as arguments for `\hspace{}`:

```
\newlength{\length}  
\settowidth{\length}{A few words}  
  
:  
  
\hspace{\length}
```

Lengths can be modified by multiplication:

```
\includegraphics[width=0.5\textwidth]{figure
```

Helpful lengths are e. g.

`\textwidth` width of the text

`\textheight` heights of the text

`\colwidth` width of a column when defined

`\baselineskip` line spacing

`\parindent` indent at the beginning of a paragraph

`\parskip` vertical distance between two paragraphs

Helpful lengths are e. g.

`\textwidth` width of the text

`\textheight` heights of the text

`\colwidth` width of a column when defined

`\baselineskip` line spacing

`\parindent` indent at the beginning of a paragraph

`\parskip` vertical distance between two paragraphs

With `\parindent` and `\parskip` you can for example customize the layout of your paragraphs.

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing**
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

In order to change the line spacing one can use the setspace package.

Example:

```
\usepackage[onehalfspacing]{setspace}
```

creates a document with 1.5 line spacing.

In order to change the line spacing one can use the setspace package.

Example:

```
\usepackage[onehalfspacing]{setspace}
```

creates a document with 1.5 line spacing.

Possible options:

`singlespacing` (standard)

`onehalfspacing`

`doublespacing`

In order to change the line spacing one can use the setspace package.

Example:

```
\usepackage[onehalfspacing]{setspace}
```

creates a document with 1.5 line spacing.

Possible options:

`singlespacing` (standard)

`onehalfspacing`

`doublespacing`

Those options may be used as a declaration (switch), e. g.

`\doublespacing` changes the format to double spacing. (from that point on).

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer**
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

The `\pagestyle{style}` command can be used to layout the content of the header and footer. `style` can be chosen from the following:

`empty` empty header and footer

`plain` page number in the footer (standard)

`headings` chapter or section titles in the header (formatting depends on the document class)

`myheadings` content of header and footer can be defined manually (the `scrlayer-scrpage` package offers an even greater variety)

If the `\pagestyle{style}` is placed within the preamble it affects the whole document, however if placed within the document it only has effect starting from the placement of the command. If you want to change the layout of a single page you can use `\thispagestyle{style}`.

With the scrlayer-scrpage package you can customize the header and the footer of the document. For further information visit:

<http://sunsite.informatik.rwth-aachen.de/ftp/pub/mirror/ctan/macros/latex/contrib/koma-script/doc/scrguien.pdf>

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page**
- 6 Multiple L^AT_EX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

If `titlepage` is set as an optional argument for the document class, a separated title page will be created. This is standard for some document classes.

If `titlepage` is set as an optional argument for the document class, a separated title page will be created. This is standard for some document classes.

Example:

```
\documentclass[titlepage]{scrartcl}
```

In order to customize your title page you can use the titlepage environment.

The title page is defined within `\begin{titlepage}... \end{titlepage}`.

Many examples may be found in the linked document.

[http://ftp.jaist.ac.jp/pub/CTAN/info/latex-samples/
TitlePages/titlepages.pdf](http://ftp.jaist.ac.jp/pub/CTAN/info/latex-samples/TitlePages/titlepages.pdf)

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files**
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

In order to structure the document you can use different files (e. g. for different chapters). You include those files into your main document using

```
\include{file/path}
```

or

```
\input{file/path}
```

The command `\input` includes the content without any page breaks, whereas `\include` causes a page break before and after the content of the loaded file. Furthermore one can not nest `\include`.

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands**
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

To define a new command you can use

```
\newcommand{\commandname}{definition of  
command}
```


To define a new command you can use

```
\newcommand{\commandname}{definition of  
command}
```

Example:

```
\newcommand{\degcelsius}{\,^{\circ}\mathrm{C}}
```

To define a new command you can use

```
\newcommand{\commandname}{definition of  
command}
```

Example:

```
\newcommand{\degcelsius}{\,\text{\textcircled{\mathrm{C}}}}
```

From now on you can use `\degcelsius` throughout your whole document:
`$T=23\degcelsius$` creates $T = 23^{\circ}\text{C}$.

You can also define commands with arguments:

```
\newcommand{\commandname}[number of  
arguments]{definition of command}
```

You can also define commands with arguments:

```
\newcommand{\commandname}[number of  
arguments]{definition of command}
```

Example:

```
\newcommand{\e}[1]{\ensuremath{\times  
10^{\#1}}}
```

You can also define commands with arguments:

```
\newcommand{\commandname}[number of  
arguments]{definition of command}
```

Example:

```
\newcommand{\e}[1]{\ensuremath{\times  
10^{\#1}}}
```

Now `\e{}` can be used as command: `$N_A=6.022\e{23}$` creates $N_A = 6.022 \times 10^{23}$.

In order to redefine an existing command you have to use `\renewcommand{} [] {}` instead of `\newcommand{} [] {}` with the same syntax.

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography**
- 9 Bibtex
- 10 Biblatex
- 11 Tikz

In order to create a bibliography you can use the thebibliography environment.

Example:

```
\begin{thebibliography}{2}
  \bibitem{jackson} J.D. Jackson:
    \emph{Classical Electrodynamics.}
    $3^{\mathrm{rd}}$ Ed., New York: Wiley, 1999.
  \bibitem{sakurai} J.J. Sakurai:
    \emph{Advanced Quantum Mechanics.}
    Reading, Mass.: Addison-Wesley, 1967.
\end{thebibliography}
```

The second argument {2} denotes the longest label in the bibliography.



J.D. Jackson: *Classical Electrodynamics*. 3rd Ed., New York: Wiley, 1999.



J.J. Sakurai: *Advanced Quantum Mechanics*. Reading, Mass.: Addison-Wesley, 1967.

The bibliography items can be referenced in the text using `\cite{key}`, which creates a label based in the definition in the bibliography.

Example:

```
Further calculations may be found in the textbook  
of J.D. Jackson \cite{jackson}.
```

creates

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex**
- 10 Biblatex
- 11 Tikz

The Bibtex packages offers the possibility to create a bibliography based on a `.bib` file.

The Bibtex packages offers the possibility to create a bibliography based on a .bib file.

The entries of a .bib file look for example like this:

```
@article{rechtsman2013PhoFlotopins ,  
  author = {Rechtsman, Mikael C and Zeuner, Julia M and  
    Plotnik, Yonatan and Lumer, Yaakov and Podolsky,  
    Daniel and Dreisow, Felix and Nolte, Stefan and  
    Segev, Mordechai and Szameit, Alexander},  
  journal = {Nature},  
  number = {7444},  
  pages = {196--200},  
  publisher = {Nature Publishing Group},  
  title = {{Photonic Floquet topological insulators}},  
  url = {http://www.nature.com/nature/journal/v496/n7444/  
    abs/nature12066.html},  
  volume = {496},  
  year = {2013}  
}
```

```
\usepackage{hyperref}
```

```
\begin{document}
```

```
    As it was shown by Rechtsman et al.
```

```
        \cite{rechtsman2013PhoFlotopins} \dots
```

```
\bibliography{literature}
```

```
\bibliographystyle{ieeetr}
```

```
\end{document}
```

```
\usepackage{hyperref}
```

```
\begin{document}
```

```
    As it was shown by Rechtsman et al.
```

```
        \cite{rechtsman2013PhoFlotopins} \dots
```

```
\bibliography{literature}
```

```
\bibliographystyle{ieeetr}
```

```
\end{document}
```

Styles include:

```
abbrv, acm, alpha, apalike, ieeetr, plain, siam, unsrt
```

The compilation here is a little tricky. We need to compile the bibliography
Along with the document! *twice!*

```
pdflatex file  
bibtex file  
pdflatex file  
pdflatex file
```

Sorry, that's just how it goes.

- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex**
- 11 Tikz

```
\usepackage{hyperref}

\usepackage[sorting=none, hyperref, maxbibnames=6]{biblatex}

\addbibresource{literature.bib}

\begin{document}

    As it was shown by Rechtsman et al.
        \cite{rechtsman2013PhoFlotopins} \dots

    \printbibliography

\end{document}
```

Like last time, it's a little silly...

For biber:

```
pdflatex file  
biber file  
pdflatex file  
pdflatex file
```

For bibtex:

```
pdflatex file  
bibtex file  
pdflatex file  
pdflatex file
```

Like last time, it's a little silly...

For biber:

```
pdflatex file  
biber file  
pdflatex file  
pdflatex file
```

For bibtex:

```
pdflatex file  
bibtex file  
pdflatex file  
pdflatex file
```

Additionally, for bibtex:

```
\usepackage[sorting=none, hyperref,  
    maxbibnames=6, backend=bibtex]{biblatex}
```

- [1] Urna Basu, Valerio Volpati, Sergio Caracciolo, and Andrea Gambassi. Short-time behavior and criticality of driven lattice gases. *Phys. Rev. Lett.*, 118:050602, Feb 2017.

One of the advantages of the biblatex package are the numerous options.

Documentation: <http://ftp.jaist.ac.jp/pub/CTAN/macros/latex/contrib/biblatex/doc/biblatex.pdf>

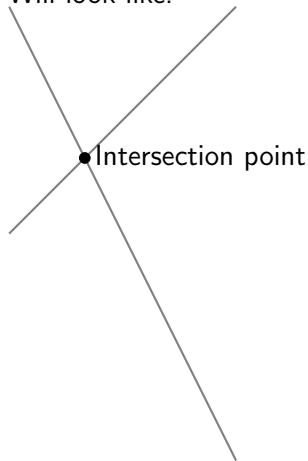
- 1 Units and Distances
- 2 Counters and Lengths
- 3 Line Spacing
- 4 Header and Footer
- 5 Title Page
- 6 Multiple \LaTeX Files
- 7 New Commands
- 8 Bibliography
- 9 Bibtex
- 10 Biblatex
- 11 Tikz**

Now, I know what you're thinking: *Man, I wish I knew how to draw things in \LaTeX !* Well, you are in luck!

```
\begin{tikzpicture}
\draw[gray, thick] (-1,2) -- (2,-4);
\draw[gray, thick] (-1,-1) -- (2,2);
\filldraw[black] (0,0) circle (2pt)
    node[anchor=west] {Intersection point};

\end{tikzpicture}
```

Will look like:



There are plenty of interesting things you can draw with tikz and I encourage you to look it up!

- There are plenty of cool things to do with \LaTeX so you can make your document look however you like!
- We covered units, spacing, headers and footers, multiple files, new commands, bibliographies, and tikz!
- Next time, we'll use real templates to do real things!