



The T_EX Typesetting System

Jun Ma

majun@nju.edu.cn

What is Typesetting?

Typesetting is the composition of text by means of arranging physical type (or sort) in *mechanical systems* or glyphs in *digital systems* representing characters (letters and other symbols)

A Brief "History" of Typesetting ()

Without Printing ...



Manual Transcription

Woodblock Printing ()



Diamond Sutra scroll (868 AD).

Movable-Type Printing ()



(970–1051 AD).



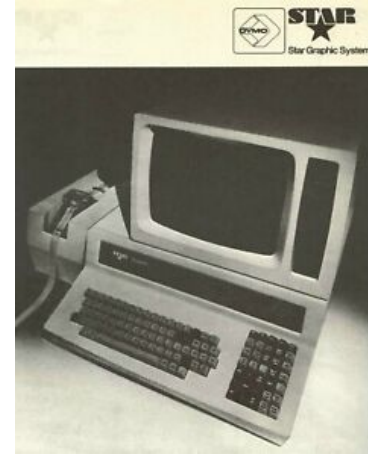
Clay movable type



Then ...



Phototypesetting (introduced in 1949)



Star Vintage Computer Typesetting machine (1973)

Now ...



TEX

Donald E. Knuth

() "father of the analysis of algorithms"



TAOCP

For his major contributions to the *analysis of algorithms and the design of programming languages*, and in particular for his contributions to "The Art of Computer Programming"...

The **T**_EXbook

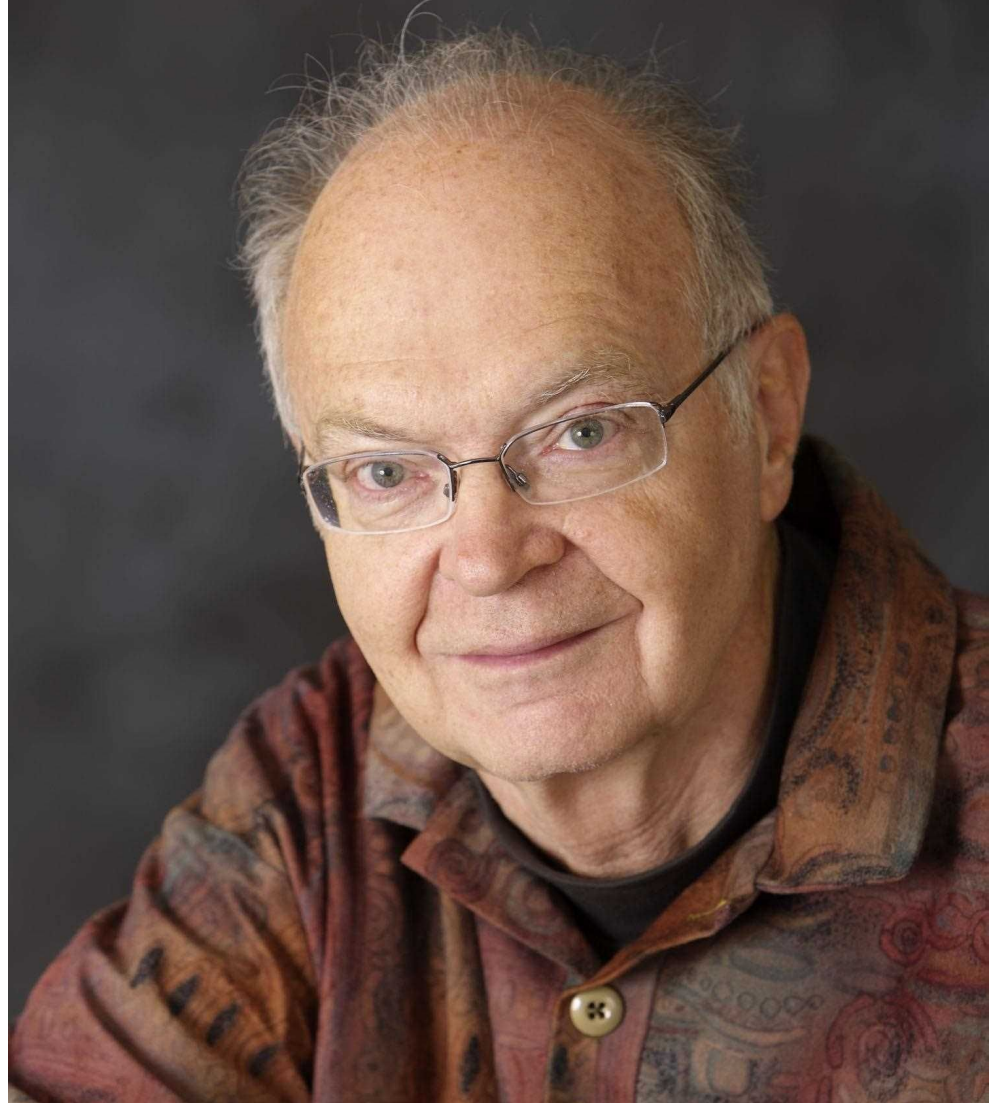
DONALD E. KNUTH Stanford University

Illustration by
MICHAEL SIBNEY



AMERICAN MATHEMATICAL
SOCIETY
1000 G STREET, N.W.
WASHINGTON, D.C. 20005
TEL: 202-338-8500
WWW.AMS-ONLINE.ORG

Intended for the creation of *beautiful* books – and especially for books that contain a lot of *mathematics*.



What is T_EX / L_AT_EX?

The T_EXbook

DONALD E. KNUTH *Stanford University*

Illustrations by
DUANE BIBBY



ADDISON-WESLEY
PUBLISHING COMPANY
Boston, Massachusetts
San Francisco · New York
Toronto · Montréal
London · Munich
Paris · Madrid
Capetown · Sydney · Tokyo
Singapore · Mexico City

- A high-level programming language (and toolchain) for typesetting
 - produces a lower-level language
- Less user-friendly
 - v.s. WYSIWYG (e.g., Microsoft Word)
- Better flexibility

- T_EX (by Knuth in 1978)
 - Commands are quite basic
- L_AT_EX (by Leslie Lamport in 1986)
 - A higher level language.
 - A set of commands defined in terms of the underlying TeX commands
 - E.g., `\usepackage{...}`, `\documentclass{...}`

Use L^AT_EX now~

Cloud editors tex.nju.edu.cn or overleaf.com

- Supports Vim/Emacs key bindings!
- Supports **co-editing**
 - More convenient than git
- Let's do it!

Binary

- [Texlive binary](#) (Windows/macOS/Linux)
 - Binary install package
 - `\apt install texlive-full` (Linux)`
- [MikTeX binary](#) (Windows/macOS/Linux)
 - Binary install package
 - `\apt install miktex` (Linux)`

Producing High Quality Documents

- Characters, words, formulas, paragraphs
- Figures and plots
- Tables

Characters

Fonts

- Sans Serif vs. Serif



- Sans Serif Fonts

- Helvetica/Arial/Microsoft YaHei
- Presentation

- Serif Fonts

- Times/Georgia/SimSun
- Documents

- Documentary: **Helvetica**

- In the Web Age: Google Fonts

- Widely used fonts

- acmart: Linux Libertine



- L^AT_EX Default: Computer Modern Roman (CMR)



Font Modification

When to use different fonts in L^AT_EX?

- `\emph{}`: *italics*
- `\textit{}`: *italics*
- `\textbf{}`: **bold**
- `\texttt{}`: typewriter
- `\textsf{}`: sans-serif
- `\textsc{}`: CAPITAL

- `\emph{}`, `\textit{}`,
`\textbf{}`
 - emphasis
- `\texttt{}`
 - code, var
 - url
- `\textsc{}`
 - SMALL CAPS
 - name of the tool:
TOOLNAME: ...

Words

Kerning and Ligature ()

TEX system has nearly perfect kerning

- The TeXbook
- Example:

ff yields ff; fi yields fi; fl yields fl; ffi yields ffi; ffl yields ffl;

Monospaced/Fixed-Width fonts

- Fira Code
- Cascadia Code
- ...

The image compares two monospaced fonts: Fira Code (v1.204) and Fira Mono. Fira Code is annotated with various features:

- equality**: Symbols like `==`, `≠`, `≡`, `≠`, `≡`.
- decent arrows**: Symbols like `<<`, `<`, `>`, `>>`, `<=>`, `>=<`, `<~>`, `>~<`.
- # pipes**: Symbols like `<<<`, `<<`, `<`, `>`, `>>`, `>>>`.
- askell stuff**: Symbols like `<|`, `<|`, `<|`, `<|`, `<|`, `<|`, `<|`.
- comparisons**: Symbols like `<=`, `<`, `>`, `>=`, `>`, `>`, `>`.
- refined decimal x**: Symbols like `\|`, `\|`, `/*`, `*/`, `///`, `//`.
- fine-tuned spacing**: Symbols like `</`, `<!--`, `</>`, `-->`, `/>`.
- textual-align**: Symbols like `0xFF`, `10x10`, `9:45`, `m-x`, `m+x`, `*ptr`.
- safe navigation operators**: Symbols like `;;`, `::`, `:::`, `..`, `...`, `..<`, `!!`, `??`, `%%`, `&&`, `||`, `?.`, `?:`.
- enlarged arithmetic**: Symbols like `+`, `++`, `+++`, `-`, `--`, `---`, `*`, `**`, `***`.
- world wide web ligature**: Symbols like `≈`, `≈`, `www`, `~`, `~@`.
- shebang**: Symbols like `#!`, `##`, `###`, `####`.
- markdown headers**: Symbols like `#[`, `#[`, `#[`, `#[`, `#?`, `#_`, `#_`.
- overline**: Symbols like `<`, `<`, `h`, `h`, `h`, `>`, `>`.

Fira Mono is shown as a baseline font without these annotations. The annotations highlight the superior kerning and ligature support in Fira Code.

Dashes, Hyphens, and Minus Signs

- **Hyphens:** for compound words like ‘daughter-in-law’ and ‘X-rated’.
- **En-dashes:** for number ranges like ‘pages 13–34’, and also in contexts like ‘exercise 1.2.6–52’.
- **Em-dashes:** for punctuation in sentences—they are what we often call simply dashes.
- **Minus:** in formulas.
 - for a hyphen (–), type a hyphen (-);
 - for an en-dash (—), type two hyphens (--);
 - for an em-dash (—), type three hyphens (---);
 - for a minus sign (−), type a hyphen in mathematics mode (\$-\$).

Formulas

TEX is for Scientists

Beautiful formulas

$$\begin{aligned}\left(\int_{-\infty}^{\infty} e^{-x^2} dx\right)^2 &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-(x^2+y^2)} dx dy \\ &= \int_0^{2\pi} \int_0^{\infty} e^{-r^2} r dr d\theta \\ &= \int_0^{2\pi} \left(-\frac{e^{-r^2}}{2} \Big|_{r=0}^{r=\infty}\right) d\theta \\ &= \pi.\end{aligned}$$

- Vedio: *Behind TEX and formula*

(Needs a lot of tuning)

- E.g., `\frac{}` VS `\cfrac{}`

```
1  $$
2  \frac{\frac{a}{b}}{c-d}
3  $$
```

$$\frac{\frac{a}{b}}{c-d}$$

```
1  $$
2  \cfrac{\cfrac{a}{b}}{c-d}
3  $$
```

$$\cfrac{\cfrac{a}{b}}{c-d}$$

Follow Conventions

- $History.append(nextState(currentState));$
- $H \leftarrow H :: \delta(\sigma)$
- Follow the natural meanings of symbols:
 - $\alpha, \beta, \dots, \Delta, \Phi, \dots$
 - $M, t, \mathbf{X}, \mathbb{N}, \mathcal{F}, \dots$
 - \prec, \succ, \dots
- texdoc symbols

Paragraphs

Words Breaks

Words may not properly break

- Most of the time, use `\-`
- `\hyphenation` or `\babelhyphenation` (provided by Babel) may be useful
- Search the friendly Web

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. 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. **ThisIsALongLongLongLongLongLongLongVariableName**

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. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. **ThisIsALongLongLong-LongLongLongLongVariableName**

Ties (~)

Same as space, except that $\text{T}_{\text{E}}\text{X}$ won't break a line at this space.

- In references to named parts of a document:

`\Chapter~12` `\Theorem~1.2` `\Appendix~A` `\Table~\hbox{B-8}` `\Figure~3` `\Lemmas 5 and~6`

- Between a person's forenames and between multiple surnames:

`\Donald~E. Knuth` `\Luis~I. Trabb~Pardo` `\Bartel~Leendert` `\van~der~Waerden` `Charles~XII`

- Between math symbols in apposition with nouns:

`\dimension~d` `\width~w` `\function~$f(x)$` `\string~s of length~l`

- When cases are being enumerated within a paragraph:

`\(b)~Show that $f(x)$ is (1)~continuous; (2)~bounded.`

- More (refer to the TeXBook)

Force break (`\break`)

Force TEX to break between lines at a certain point in the middle of a paragraph

- 1 If you want to force `\TeX\xspace` to break between lines at a certain point in the middle of a paragraph, just say
- 2 However, that might cause the line to be really spaced out.`\break`
- 3 If you want `\TeX\xspace` to fill up the right-hand part of a line with blank space just before a forced line break,
- 4 without indenting the next line, say '`\backslashhfil\backslashbreak`'.

If you want to force TEX to break between lines at a certain point in the middle of a paragraph, just say '`\break`'. However, that might cause the line to be really spaced out.

If you want `TEX` to fill up the right-hand part of a line with blank space just before a forced line break, without indenting the next line, say '`\hfil\break`'.

For Those Picky Guys

Use ``microtype``!

- Subliminal refinements towards typographical perfection

Microtype is ON

Transport processes in porous media occur in many diverse fields of science and engineering, including groundwater pollution, oil recovery, chromatographic separations, filtration, drug discovery and peptide engineering, evolution of rocks, diffusion in biological tissues, and many others. Accurate

LaTeX Typesetting: Fun Facts

It's an iterative algorithm

- Each iteration leaves some results (.aux, .bbl, ...)
- Next iteration uses these results
 - e.g., a fresh build has “??” and no references

LaTeX does not guarantee a fixed point

- Table of contents (your thesis)
 - blank → correct TOC → correct TOC pages
- There can be documents without a fixed point

Figures and Tables

Why Use Figures?

- Help readers understand the flow of (dense) technical contents

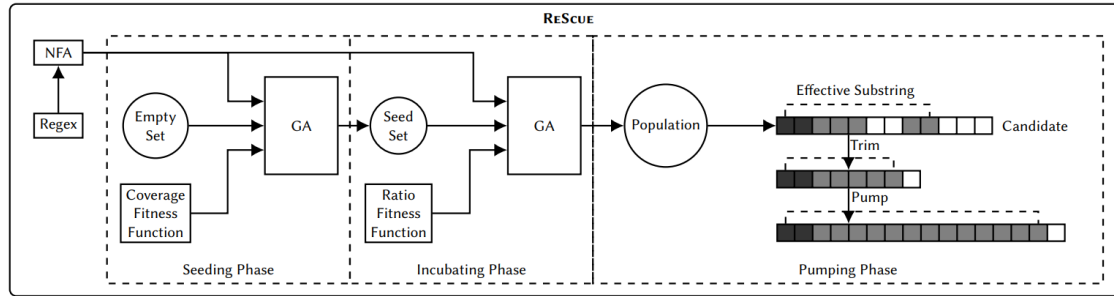


Figure 1: Overview of the ReSCUE technique for automated ReDoS string generation.

- Show the evaluation results

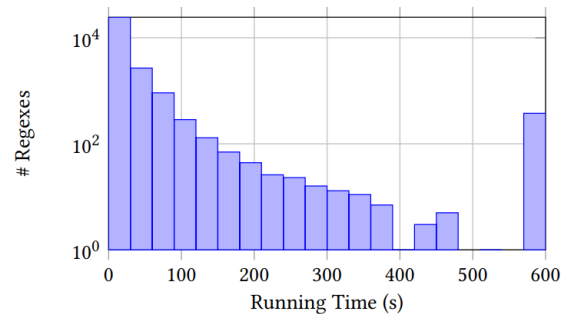


Figure 3: The histogram of ReSCUE runtime over all evaluated regexes

Tips for figures

Be informative

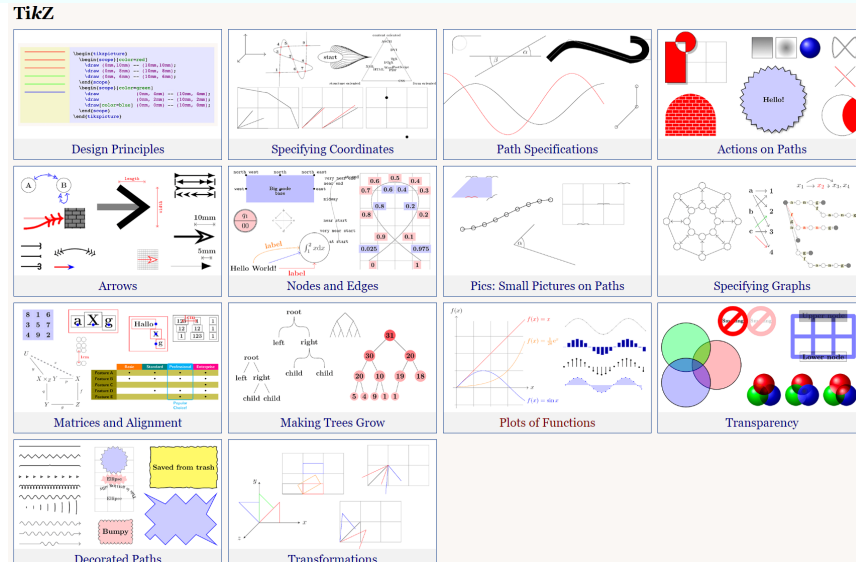
- don't just list a few boxes
- explain (in text/caption) this well

Be self-contained

- figures + captions = everything

Drawing Graphics with TiKZ & PGF

Basically, TiKZ just defines a number of TEX commands that draw graphics



Perfectly integrates with the TeX system

- Example: connecting two document parts

Tables

Tables are generally used for showing, e.g., experimental subjects, results ...

- Tips for tables
 - Be readable
 - use visual signals to help readers identify your message
 - Be self-contained
 - same as figures
 - (minor) Fine tune your tables
 - ComboDroid (ICSE'20)

Table 1: Evaluation results of ComboDroid^α: test coverage

Subject (Category, Downloads; LoC)	Monkey	Sapienz	APE	ComboDroid ^α	Coverage trend
WORDPRESS, WP (Social, 5M-10M; 327,845)	24.4%	24.3%	24.1%	36.1% (+11.7%)	
ANTENNAPOD, AP (Video, 100K-500K; 262,460)	57.5%	61.3%	65.5%	69.8% (+4.3%)	
K-9 MAIL, K9 (Communication, 5M-10M; 159,708)	19.1%	20.4%	26.3%	32.5% (+6.2%)	
MYEXPENSES, ME (Finance, 500K-1M; 104,306)	43.8%	40.2%	48.6%	56.3% (+7.7%)	
WIKIPEDIA, WIKI (Books, 10M-50M; 93,404)	37.2%	39.3%	44.3%	45.1% (+0.8%)	
ANKIDROID, AD (Education, 1M-5M; 66,513)	50.6%	49.0%	50.6%	54.3% (+3.7%)	
AMAZEFILEMANAGER, AFM (Tools, 100K-500K; 66,126)	39.6%	42.5%	45.0%	53.2% (+10.2%)	
POCKETHUB, PH (Tools, 100K-500K; 47,946)	22.1%	19.1%	27.2%	31.4% (+4.2%)	
ANYMEMO, AM (Education, 100K-500K; 40,503)	57.5%	51.7%	64.3%	66.8% (+2.5%)	
HACKER NEWS READER, HNR (News, 50K-100K; 38,315)	69.9%	66.2%	65.5%	71.2% (+1.3%)	
CALLMETER, CM (Tools, 1M-5M; 21,973)	54.0%	49.1%	58.5%	60.4% (+1.9%)	
SIMPLETASK, ST (Productivity, 10K-50K; 20,980)	57.2%	57.2%	62.8%	70.2% (+7.4%)	
SIMPLE DRAW, SD (Tools, 10K-50K; 18,685)	50.0%	51.3%	22.8%	26.8% (-24.5%)	
AARD2, AARD (Books, 10K-50K; 9,622)	68.0%	64.3%	73.8%	77.6% (+3.8%)	
WORLD CLOCK, WC (Bussiness, 1M-5M; 7,181)	50.2%	50.8%	55.1%	58.0% (+2.9%)	
COOLCLOCK, CC (Tools, 10K-50K; 2,762)	75.4%	73.2%	78.0%	79.6% (+1.6%)	
ALOGCAT, ALC (Tools, 100K-500K; 846)	49.1%	48.8%	49.1%	49.1% (0.0%)	
Average	48.6%	47.6%	50.7%	55.3% (+4.6%)	

¹ Column **Coverage trend** plots the coverage trend of each tool. The red solid lines denote ComboDroid^α, and dashed lines are existing techniques. The detailed coverage trends are displayed in Figure 2. Number in a bracket is the coverage difference between ComboDroid^α and the best existing technique (Monkey, Sapienz, and APE).

Table 2: Evaluation results of ComboDroid^α: bug manifestation

Bug ID	Cause	Discovered by
WP-10147	Infinite recursion	APE, CD ^α
AP-1234	Atomicity violation	CD ^α
AP-3195	Null pointer dereference	all
K9-3308	Mismatched mime type	Sapienz, CD ^α
AFM-1351	Null pointer dereference	all
AFM-1402	Lifecycle event mishandling	APE, CD ^α
AM-480*	Lifecycle event mishandling	CD ^α
AM-503*	Null pointer dereference	APE, CD ^α
CM-128*	Text input mishandling	APE, CD ^α
SD-49	Miss-used local variables	Monkey
AARD-90*	Null pointer dereference	CD ^α
AARD-7	Null pointer dereference	all

Monkey: 4 (33%); Sapienz: 4 (33%); APE: 7 (58%); CD^α: 11 (92%)

¹ Bug ID is the issue ID in the project's GitHub repository. A starred Bug ID* denotes a previously unknown bug.

Table 3: Evaluation results of ComboDroid^β: test coverage

Subject	UC	CD ^α	ComboDroid ^β	Expert
WP (328K)	40.1%	36.1%	48.3% (+8.2%/+12.2%)	53.3% (+5.0%)
AP (262K)	65.4%	69.8%	75.2% (+9.8%/+5.4%)	78.6% (+3.4%)
K9 (160K)	38.5%	32.5%	50.3% (+11.8%/+17.8%)	53.7% (+3.4%)
ME (104K)	53.1%	56.3%	66.8% (+13.7%/+10.5%)	69.7% (+2.9%)
WIKI (93K)	37.3%	45.1%	46.0% (+8.7%/+0.9%)	49.6% (+3.6%)
AD (67K)	50.3%	54.3%	66.8% (+16.5%/+12.5%)	71.4% (+4.6%)
AFM (66K)	43.3%	55.2%	66.2% (+22.9%/+11%)	67.3% (+1.1%)
PH (48K)	31.5%	31.4%	39.2% (+7.7%/+7.8%)	45.3% (+6.1%)
AM (41K)	62.1%	66.8%	71.3% (+9.2%/+4.5%)	74.0% (+2.7%)
HNR (38K)	53.4%	71.2%	76.5% (+23.1%/+5.3%)	76.3% (-0.2%)
Average	47.5%	51.9%	60.7% (+13.2%/+8.8%)	63.9% (+3.2%)

¹ The number in Column **Subject** is the app's LoC. Columns **UC**, **CD^α**, **ComboDroid^β**, and **Expert** display the code coverage of manual use cases, ComboDroid^α, ComboDroid^β, and the human expert, respectively. The numbers in the brackets of Column **ComboDroid^β** indicate the coverage differences between ComboDroid^β and manual use cases and ComboDroid^α, respectively. The numbers in the brackets of Column **Expert** indicate the differences between the human expert and ComboDroid^β.

Summary and More Tips

L^AT_EX is a Programming Language

Write readable code

- Split code into (well-organized) files
- Use macros
 - E.g., tool name, recall/precision/F1-Score, ...

```
1 \newcommand{\tool}{\xspace{ToolName}\xspace}
```

Host your code on Overleaf/Tex@NJU/Git

- Collaborate with your teammates
- Split long sentences into lines (to avoid collisions)

There is LaTeX diff

- See how your advisor revise your document

Generate Plain Text

Text is useful to other tools (e.g., Grammarly)

```
1  pandoc \ # use another compiler
2    --wrap=none main.tex -t plain | \
3    python3 run.py > main.txt
```

```
1  # run.py
2  import sys, re
3  print(re.sub(r"\s+([,.])", r"\1", \
4         re.sub(r"(\s)+", r"\1", \
5         sys.stdin.read())))
```

Other Useful Tricks

Command-line tools

- texdoc - manpages
- pdftocrop - very useful when your figures are PowerPoint
 - PowerPoint has COM API for automation

Packages

- microtype
- xpsace - `\compare \tool{} with...` \Rightarrow `\compare \tool with...`
- cleveref - `Figure~\ref{fig:example}` \Rightarrow `\Cref{fig:example}`
- Any more?