TEX

Linux Users Group

David Chiang

Table of contents

What?

A programming language whose programs output *documents*.

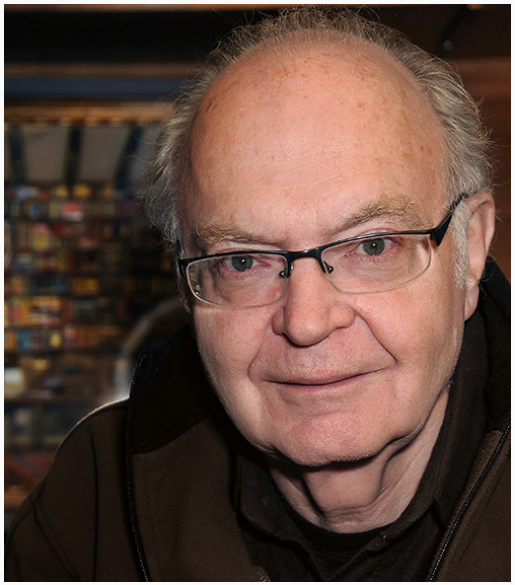
- LaTeX: A *format* used on top of (plain) TeX; by far, the most commonly used.
- pdf(La)TeX: An alternative implementation that directly generates PDF files (very common, maybe more than the original)
- Xe(La)TeX: Another alternative implementation with native support for Unicode and OpenType fonts (also very common)

TEX and Unix

- Not developed under Unix
- Not the original Unix typesetter – that honor belongs to `troff`

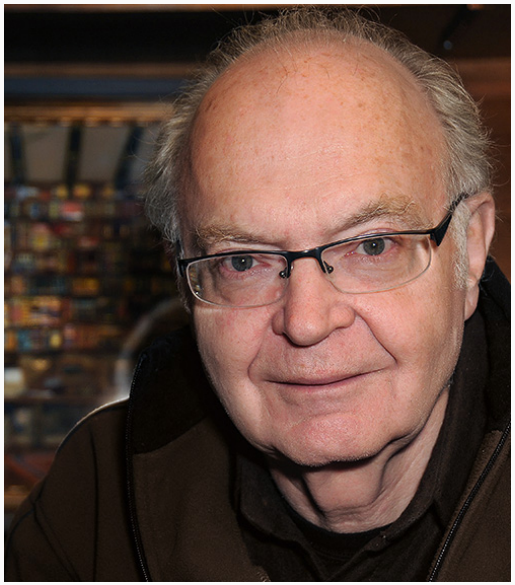
  ```
  cat file | pic | tbl | eqn | troff | lpr
  ```
- But as an open-source, command-line alternative to word processors, it has a long history with Unix/Linux

Whence?



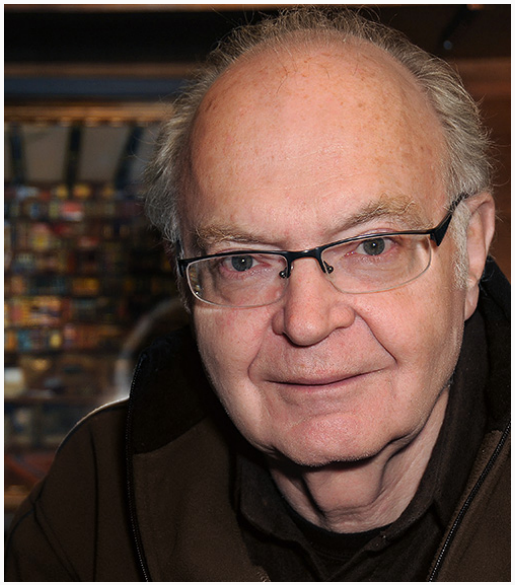
Donald Knuth

- Trying to write *The Art of Computer Programming*, vol. 2, 2nd ed.
- The publisher's new typesetter was bad
  - So he wrote his own software (TeX)
  - And two books about it



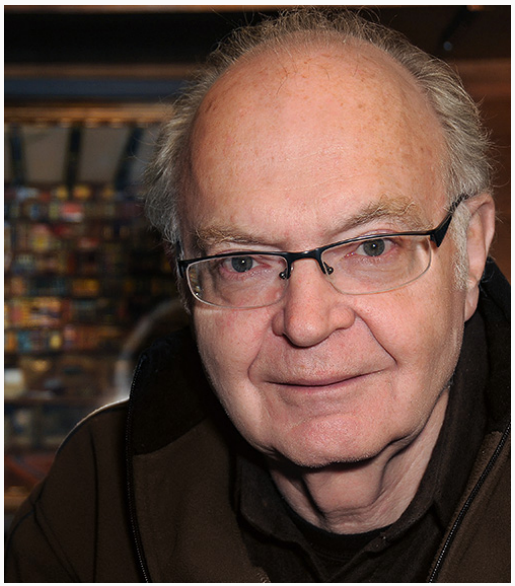
Donald Knuth

- Trying to write *The Art of Computer Programming*, vol. 2, 2nd ed.
- The publisher's new typesetter was bad
  - So he wrote his own software (TeX)
  - And two books about it
- TeX needed digital fonts
  - So he wrote his own font renderer (Metafont)
  - And two books about it
  - And designed the fonts (Computer Modern)
  - And wrote a book about that



Donald Knuth

- Trying to write *The Art of Computer Programming*, vol. 2, 2nd ed.
- The publisher's new typesetter was bad
  - So he wrote his own software (TeX)
  - And two books about it
- TeX needed digital fonts
  - So he wrote his own font renderer (Metafont)
  - And two books about it
  - And designed the fonts (Computer Modern)
  - And wrote a book about that
- Also, he invented a new programming paradigm (literate programming)
  - And wrote a book about that



Donald Knuth

- Trying to write *The Art of Computer Programming*, vol. 2, 2nd ed.
- The publisher's new typesetter was bad
  - So he wrote his own software (TeX)
  - And two books about it
- TeX needed digital fonts
  - So he wrote his own font renderer (Metafont)
  - And two books about it
  - And designed the fonts (Computer Modern)
  - And wrote a book about that
- Also, he invented a new programming paradigm (literate programming)
  - And wrote a book about that
- Meanwhile, *TAOCP* is still not finished

- Written in WEB, a mix of Pascal and TEX
- Frozen in 1989 at version 3.0
- Anyone who finds a bug receives $327.68 (no longer real)

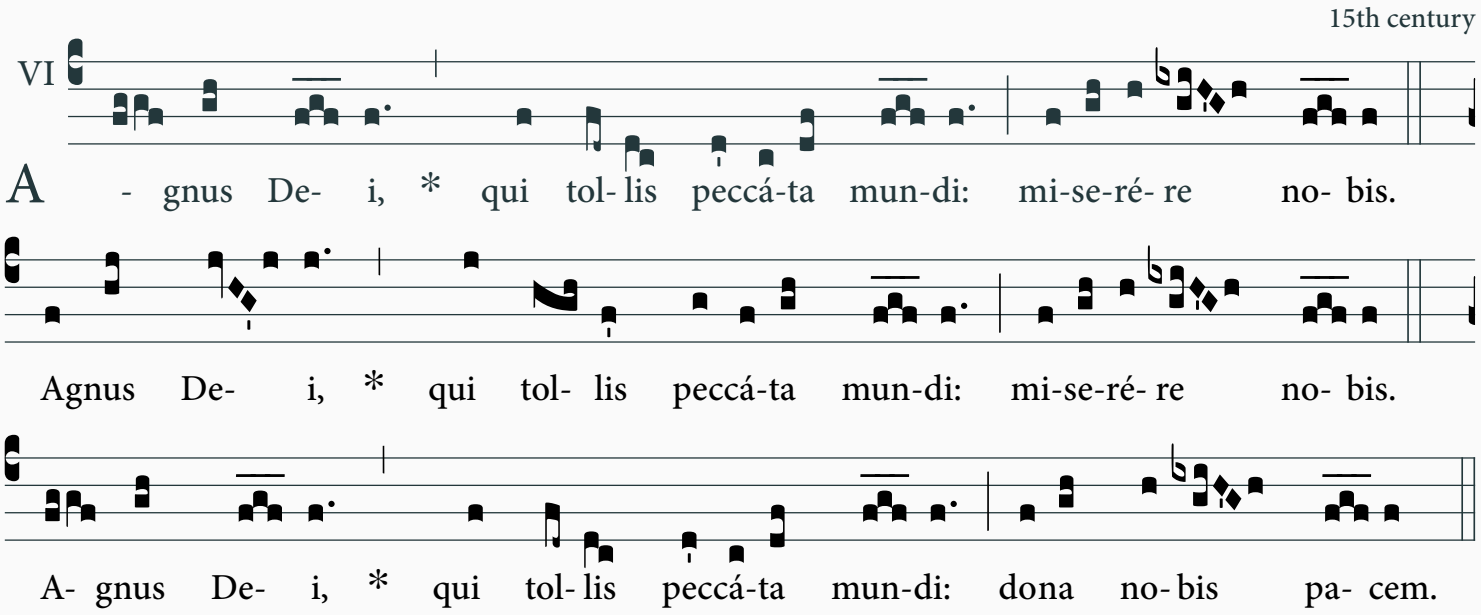
```
% This program is copyright (C) 1982 by D. E. Knuth; all rights are reserved.
% Copying of this file is authorized only if (1) you are D. E. Knuth, or if
% (2) you make absolutely no changes to your copy. (The WEB system provides
% for alterations via an auxiliary file; the master file should stay intact.)
% See Appendix H of the WEB manual for hints on how to install this program.
% And see Appendix A of the TRIP manual for details about how to validate it.

% TeX is a trademark of the American Mathematical Society.
% METAFONT is a trademark of Addison-Wesley Publishing Company.

% Version 0 was released in September 1982 after it passed a variety of tests.
% Version 1 was released in November 1983 after thorough testing.
% Version 1.1 fixed ``disappearing font identifiers'' et alia (July 1984).
% Version 1.2 allowed `0' in response to an error, et alia (October 1984).
% Version 1.3 made memory allocation more flexible and local (November 1984).
% Version 1.4 fixed accents right after line breaks, et alia (April 1985).
% Version 1.5 fixed \the\toks after other expansion in \edefs (August 1985).
% Version 2.0 (almost identical to 1.5) corresponds to "Volume B" (April 1986).
% Version 2.1 corrected anomalies in discretionary breaks (January 1987).
% Version 2.2 corrected "(Please type...)" with null \endlinechar (April 1987).
% Version 2.3 avoided incomplete page in premature termination (August 1987).
% Version 2.4 fixed \noaligned rules in indented displays (August 1987).
% Version 2.5 saved cur_order when expanding tokens (September 1987).
% Version 2.6 added 10sp slop when shipping leaders (November 1987).
% Version 2.7 improved rounding of negative-width characters (November 1987).
% Version 2.8 fixed weird bug if no \patterns are used (December 1987).
```

Why?

Batteries included

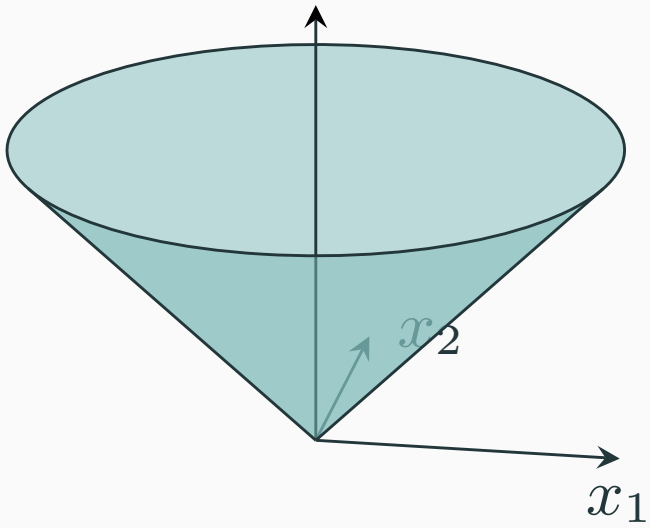
Communion 領聖體禮

Priest	The peace of the Lord be with you always.	主禮	願主的平安常與你們同在。
All	And with your spirit.	全體	也與你的心靈同在。
Priest	Let us offer each other the sign of peace.	主禮	請大家互祝平安。

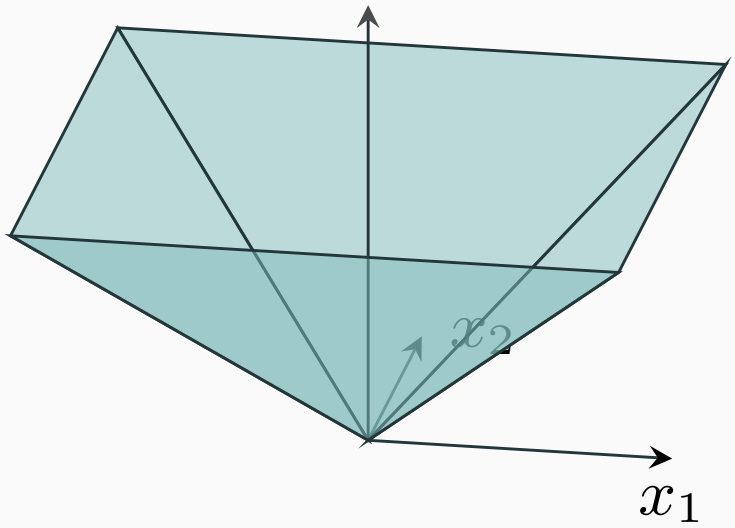



```
(c4) A(fg/gf)gnus(gh) De(f_g_f_)i,(f.) *(,)
qui(f) tol(fd~)lis(dc) pec(d')cá(c)ta(df) mun(f_g_f_)di:(f.) (;)
mi(f)se(gh)ré(h)re(ixgiH'Gh) no(f_g_f_)bis.(f) (::) (z)
```

Batteries included



ℓ_2



ℓ_∞

```
\begin{tikzpicture}[3d={70,-80}]
\draw[->] (0,0,0) -- (0,0,1.5);
...
\end{tikzpicture}
```

Programmable

Obviously, Sort-o-Matic runs in $O(n^2 \log \log n)$ time.

```
\newcommand{\algname}
           {Sort-o-Matic}
\newcommand{\bigo}[1]{O(#1)}
```

Obviously, SuperSorter runs in $\mathcal{O}(n^2 \log \log n)$ time.

```
\newcommand{\algname}
           {SuperSorter}
\newcommand{\bigo}[1]%
  {\mathcal{O}(#1)}
```

```
Obviously, \algname{} runs in
$\bigo(n^2 \log \log n)$ time.
```

Typesetting

Linux is the kernel: the program in the system that allocates the machine's resources to the other programs that you run. The kernel is an essential part of an operating system, but useless by itself; it can only function in the context of a complete operating system. Linux is normally used in combination with the GNU operating system: the whole system is basically GNU with Linux added, or GNU/Linux. All the so-called "Linux" distributions are really distributions of GNU/Linux.

TEX

Linux is the kernel: the program in the system that allocates the machine's resources to the other programs that you run. The kernel is an essential part of an operating system, but useless by itself; it can only function in the context of a complete operating system. Linux is normally used in combination with the GNU operating system: the whole system is basically GNU with Linux added, or GNU/Linux. All the so-called "Linux" distributions are really distributions of GNU/Linux.

MS Word

Theorem 4. *If M is a ring-weighted multiset automaton with d states converted from a regular expression, then*

1. $\dim \mathrm{Ins}(M) = d$.
2. $\mathrm{Ins}(M)$ *can be decomposed into a direct sum*

$$\mathrm{Ins}(M) \cong \bigoplus_{\Delta \subseteq \Sigma} \mathrm{Ins}_\Delta(M)$$

where $\mu(w) \in \mathrm{Ins}_\Delta(M)$ *iff* $\mathrm{alph}(w) = \Delta$.

Proof. By induction on the structure of the regular expression α.

If α is unary: the Cayley-Hamilton theorem gives a generating set $\{I, \mu(a), \ldots, \mu(a)^{d-1}\}$, which has size d. Moreover, let $\mathrm{Ins}_\emptyset(M)$ be the span of $\{I\}$ and $\mathrm{Ins}_{\{a\}}(M)$ be the span of the $\mu(a)^i$ $(i > 0)$. The automaton M, by construction, has a state (the initial state) with no incoming transitions. That is, its transition matrix has a zero column, which means that its characteristic polynomial has no I term. Therefore, if $w \neq \epsilon$, $\mu(w) \in \mathrm{Ins}_{\{a\}}(M)$.

If $\alpha = k\alpha_1$, then $\mathrm{Ins}(M) = \mathrm{Ins}(M_1)$, so both properties hold of $\mathrm{Ins}(M)$ if they hold of $\mathrm{Ins}(M_1)$.

If $\alpha = \alpha_1 \cup \alpha_2$, the inside weights of $M_1 \cup M_2$ for w are

$$\mu(w) = \prod_{a \in w} \mu(a) = \prod_a \begin{bmatrix} \mu_1(a) & 0 \\ 0 & \mu_2(a) \end{bmatrix} = \begin{bmatrix} \prod_a \mu_1(a) & 0 \\ 0 & \prod_a \mu_2(a) \end{bmatrix} = \begin{bmatrix} \mu_1(w) & 0 \\ 0 & \mu_2(w) \end{bmatrix}.$$

Thus, $\mathrm{Ins}(M) \cong \mathrm{Ins}(M_1) \oplus \mathrm{Ins}(M_2)$, and $\dim \mathrm{Ins}(M) = \dim \mathrm{Ins}(M_1) + \dim \mathrm{Ins}(M_2)$. Moreover, $\mathrm{Ins}_\Delta(M) \cong \mathrm{Ins}_\Delta(M_1) \oplus \mathrm{Ins}_\Delta(M_2)$.

How?

Installing locally

TeX comes in various distributions with their own package managers, and it's up to you whether to install it inside or outside your system package manager.

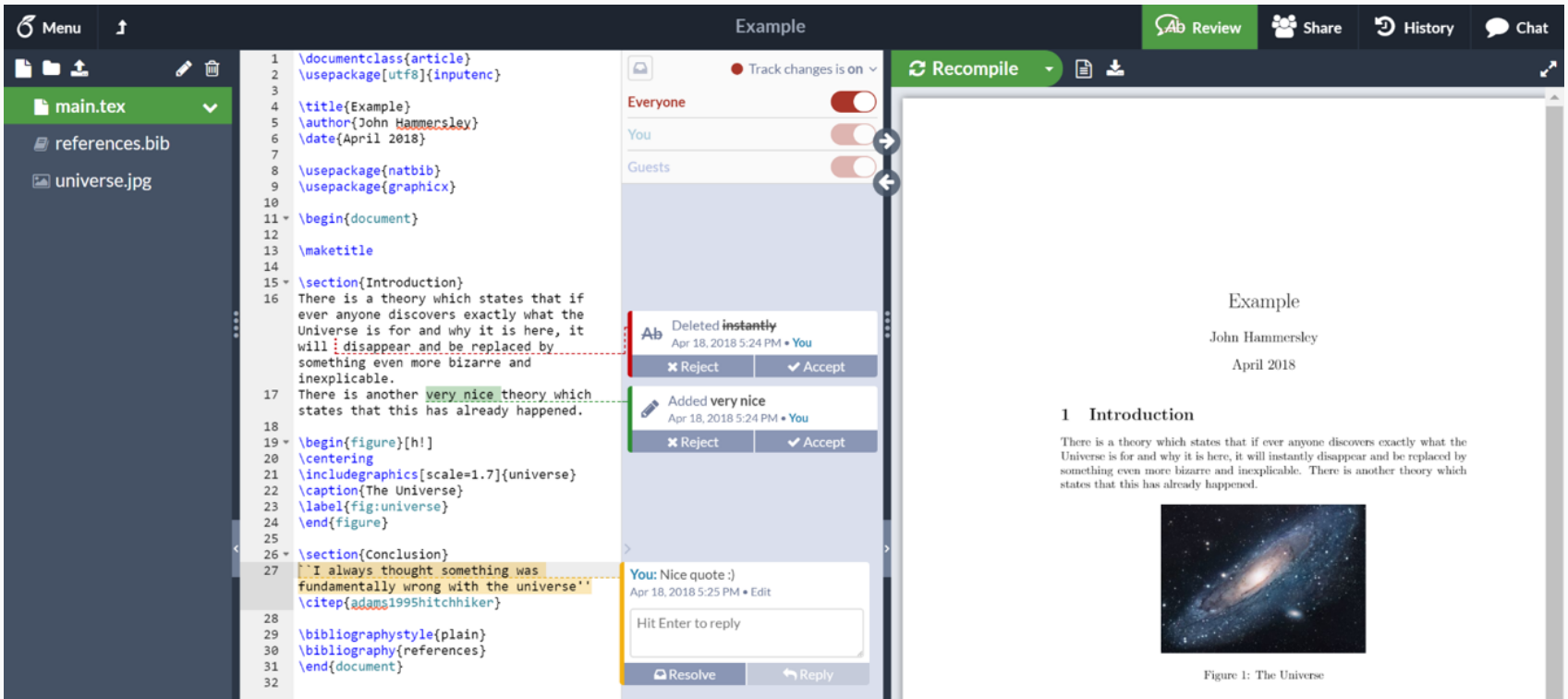
- Linux
  - manual: `https://www.tug.org/texlive/`
  - Ubuntu: `sudo apt-get texlive-full`
  - Arch: `pacman -S texlive-most texlive-lang`
- MacOS
  - manual: `http://www.tug.org/mactex/`
  - Homebrew: `brew cask install mactex`

Web-based

You can also edit and compile TEX in a web browser, which is especially good for collaborative writing.

The two most common sites are **`sharelatex.com`** and **`overleaf.com`**, which are merging into **`overleaf.com`** (so it's a little complicated right now).

Sign up with your **`nd.edu`** email address to get a free subscription.



Hello, World

```
\documentclass{article}
\begin{document}
Hello, world.
\end{document}
```

Hello, World

```
\documentclass{article}
\title{A Very Short Document}
\author{David Chiang}
\begin{document}
\maketitle
\section{Greeting}
Hello, world.
\end{document}
```

Whitespace

`a b` `a. b` `a.~b`	a b a. b a. b
`a` `b`	a b
`a` `b`	a b

Punctuation

`\$1 million`	$1 million
`100\%`	100%
`\#1`	#1
`M \& M`	M & M
`` `hi' ``	‘hi’
`` ``hi'' ```	“hi”
`574-631-9441`	574-631-9441
`1--2`	1–2
`Wait---what?`	Wait—what?

Formatting

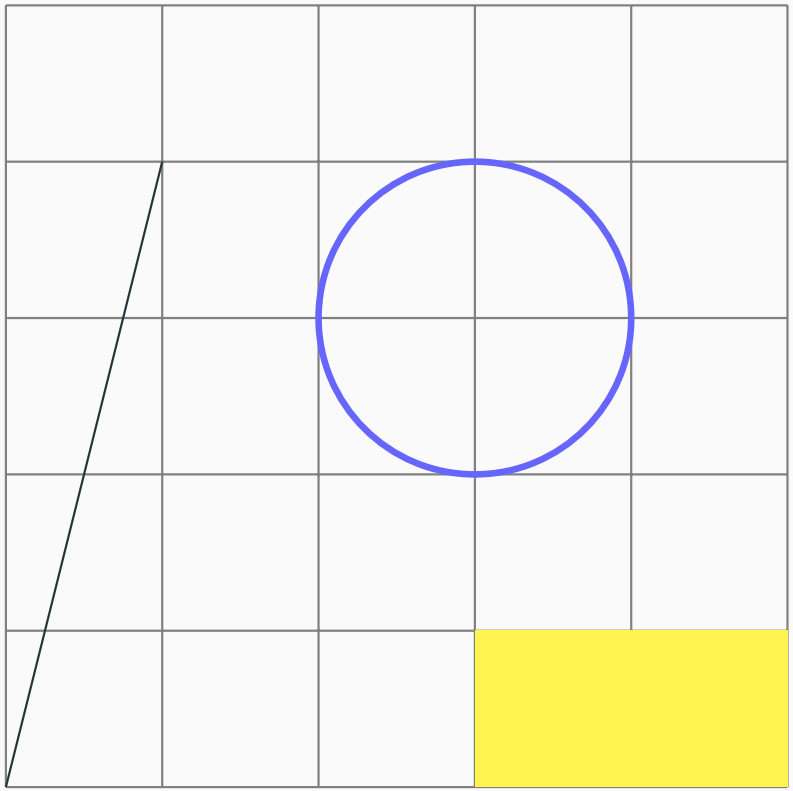
`\emph{Emphasis}`	*Emphasis*
`\textit{Italics}`	*Italics*
`\textbf{Bold}`	**Bold**
`\texttt{Typewriter}`	`Typewriter`

Pictures

- Draw it in an external program
  - In preamble: `\usepackage{graphicx}`
  - In text: `\includegraphics{mypicture.pdf}`
- Code it in the TeX source
  - TikZ is by far the most common. In preamble: `\usepackage{tikz}`
  - PSTricks is an older one, better in some ways but may require some fiddling

TikZ

```
\begin{tikzpicture}
\draw (0,0) -- (1,4);
\draw[blue!60,very thick]
  (3,3) circle[radius=1];
\fill[yellow!80]
  (3,0) rectangle (5,1);
\end{tikzpicture}
```



(Grid is normally invisible)

TikZ: paths

`\draw` *Op** `;`

where each *Op* can be (not an exhaustive list):

Coord	move to *Coord* without drawing
`--` *Coord*	straight line to *Coord*
`to` *Coord*	“user-defined” line to *Coord*
`edge` *Coord*	same, but doesn’t move
`rectangle` *Coord*	rectangle to *Coord*
`circle[radius=`*Len*`]`	circle with radius *Len*

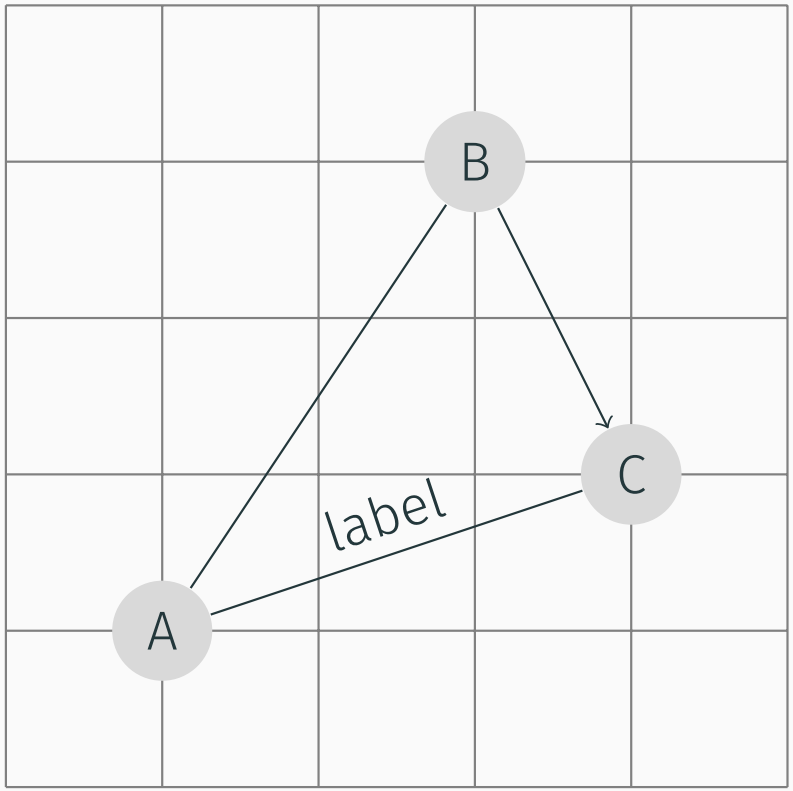
TikZ: coordinates

Coord

(*x*,*y*)	Cartesian
(*θ*:*r*)	polar
(*Name*)	node with name *Name*
++*Coord*	relative to previous point

TikZ: nodes

```
\begin{tikzpicture}
\tikzset{every node/.style=
  {fill=gray!30,circle}}
\node (a) at (1,1) {A};
\node (b) at (3,3) {B};
\draw (a) -- (b);
\draw[->] (b) -- (c);
\draw (c) -- node[edgelabel]
  {label} (a);
\end{tikzpicture}
```



(Grid is normally invisible)

TikZ: nodes

`\draw` *Op** `;`

`node (`*Name*`) {`*Text*`}`	node with name *Name* and label *Text*
`-- node {`*Text*`}` *Coord*	line with node on it
`to node {`*Text*`}` *Coord*	similar
`edge node {`*Text*`}` *Coord*	similar

`\node` is shorthand for `\draw node`.

TikZ: the `to` operation

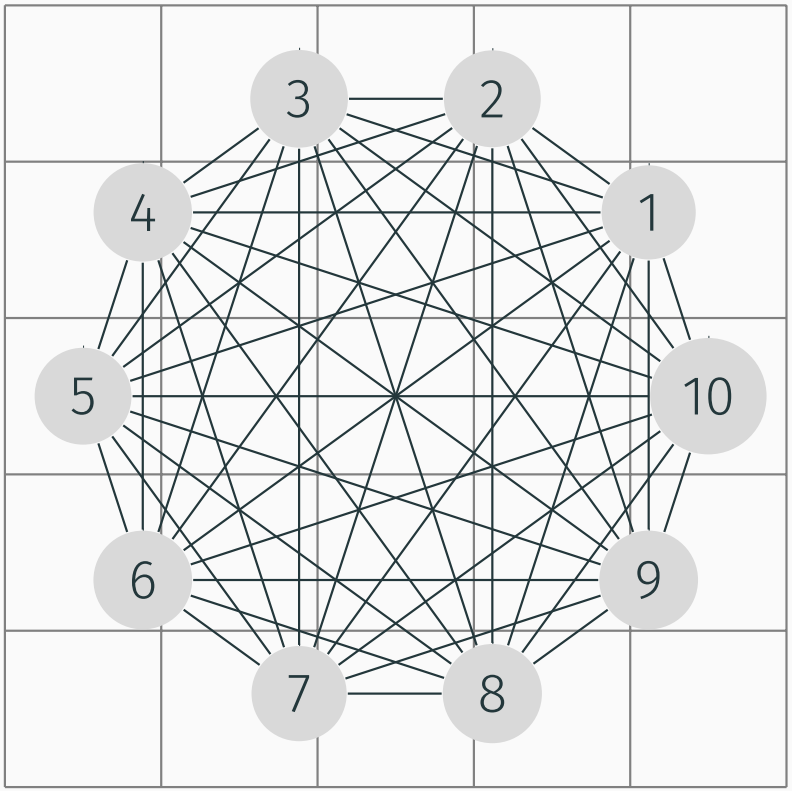
`to[`*Options*`]` *Coord*

has some extra options and is probably the most convenient way to connect nodes:

`out=`θ	line exits source node with angle θ
`in=`θ	line enters target node with angle θ
`bend left`	line goes a little to the left and curves right
`bend right`	line goes a little to the right and curves left

TikZ

```
\begin{tikzpicture}
\tikzset{every node/.style={
  fill=gray!30,circle}}
\begin{scope}[xshift=2.5cm,
              yshift=2.5cm]
\foreach \i in {1,...,6} {
  \node (n\i) at ({\i*60}:2) {\i};
}
\foreach \i in {1,...,6} {
  \foreach \j in {\i,...,6} {
    \draw (n\i) -- (n\j);
  }
}
\end{scope}
\end{tikzpicture}
```



(Grid is normally invisible)

Whither?

Further reading

- This tutorial is short:
  `http://www.lsv.fr/~schmitz/teach/2011_latex/td.pdf`
  It contains many recommendations for current best practices.
- The Overleaf documentation is more detailed:
  `https://v2.overleaf.com/learn/latex/Main_Page`
- TEX as a programming language:
  `http://eijkhout.net/texbytopic/texbytopic.html`

Important packages

Better fonts	XeLaTeX + fontspec
Better math	amsmath + amsthm + amssymb + mathtools
Better tables	booktabs
Better page layout	geometry
Bibliographies	BibTeX + natbib
Slides	beamer
Pictures	tikz
Graphs	tikz + pgfplots
Images	graphicx
Pseudocode	algorithmicx (?)