An Interactive Introduction to LaTeX

Part 3: Not Just Papers: Presentations & More

LATEX Recap

- ▶ You write your document in `plain text` with `commands` that describe its structure and meaning.
- ▶ The `latex` program processes your text and commands to produce a beautifully formatted document.

```
The rain in Spain falls \emph{mainly} on the plain.
```

latex

The rain in Spain falls *mainly* on the plain.

LATEX Recap: Commands & Arguments

- A command starts with a *backslash* [\] .
- Some commands take an *argument* in curly braces [{] [}] .
- Some commands also take *optional arguments* in square brackets [[] []] .


```
\includegraphics[
  width=0.5\textwidth]{gerbil}

\includegraphics[
  width=0.3\textwidth,
  angle=270]{gerbil}
```

Image license: CC0

LaTeX Recap: Environments

- The `\begin` and `\end` commands are used to create many different environments — contexts.
- The itemize and enumerate environments make lists.

```
\begin{itemize} % for bullet points
\item Biscuits
\item Tea
\end{itemize}

\begin{enumerate} % for numbers
\item Biscuits
\item Tea
\end{enumerate}
```

- Biscuits
- Tea

1. Biscuits
2. Tea

LaTeX Recap: Mathematics

▶ The `equation` environment makes a numbered equation.

```
\begin{equation}
  \sum_{k=1}^{n} \frac{1}{2^k}
\end{equation}
```

$$\sum_{k=1}^{n} \frac{1}{2^k} \qquad (1)$$

▶ Use dollar signs $ to mark mathematics in text.

```
% not so good:
Let a and b be distinct positive
integers, and let c = a - b + 1.

% much better:
Let $a$ and $b$ be distinct positive
integers, and let $c = a - b + 1$.
```

Let a and b be distinct positive integers, and let c = a - b + 1.

Let a and b be distinct positive integers, and let $c = a - b + 1$.

▶ Always use dollar signs in pairs — one to begin the mathematics, and one to end it.

In fact, we could have written `$...$` as `\begin{math}...\end{math}`.

LATEX Recap: Document Structure

- ▶ Starts with the `\documentclass` — what type of document.
- ▶ Metadata (`\title` and `\author`) and packages in the preamble.
- ▶ Content between `\begin{document}` and `\end{document}`.
- ▶ The `\maketitle` command creates the title; `\section` commands create numbered sections.

```latex
\documentclass{article}
% preamble
\title{The Title}
\author{A. Author}

\begin{document}
% body
\maketitle

\section{Introduction}

In this paper we \ldots

\end{document}
```

The Title

A. Author

April 30, 2013

1 Introduction

In this paper we . . .

Presentations with beamer

- Beamer is a package for creating presentations (such as this one!) in LaTeX.
- It provides the beamer document class.
- Use the frame environment to create slides.

```latex
\documentclass{beamer}

\title{Welcome to Beamer}
\author{You}
\institute{Where You're From}
\date{Date of Presentation}

\begin{document}

\begin{frame}
\titlepage % beamer's \maketitle
\end{frame}

\end{document}
```

Welcome to Beamer

You

Where You're From

Date of Presentation

Presentations with `beamer`: Frames

- ► Use `\frametitle` to give the frame a title.
- ► Then add content to the frame.
- ► The source for this frame looks like:

```
\begin{frame}
  \frametitle{Presentations with beamer: Frames}
  \begin{itemize}
    \item Use \texttt{frametitle} to give the frame a title.
    \item Then add content to the frame.
    \item The source for this frame looks like ...
  \end{itemize}
\end{frame}
```

Presentations with `beamer`: Sections

- You can use `\section`s to group your `frames`, and `beamer` will use them to create an automatic outline.
- To generate an outline, use the `\tableofcontents` command. Here's one for this presentation. The `currentsection` option highlights the current section.

`\tableofcontents[currentsection]`	LATEX Recap
	Presentations with `beamer`
	Drawings with Ti*k*Z
	Notes with `todonotes`
	Spreadsheets with `spreadtab`

Presentations with `beamer`: Multiple Columns

- ▶ Use the `columns` and `column` environments to break the slide into columns.
- ▶ The argument for each `column` determines its width.
- ▶ See also the `multicol` package, which automatically breaks your content into columns.

```
\begin{columns}
  \begin{column}{0.4\textwidth}
    \begin{itemize}
    \item Use the columns ...
    \item The argument ...
    \item See also the ...
    \end{itemize}
  \end{column}
  \begin{column}{0.6\textwidth}
    % second column
  \end{column}
\end{columns}
```

- ▶ Use `\emph` or `\alert` to highlight:

`I should \emph{emphasise} that` `this is an \alert{important} point.`	I should *emphasise* that this is an important point.

- ▶ Or specify bold face or italics:

`Text in \textbf{bold face}.` `Text in \textit{italics}.`	Text in **bold face**. Text in *italics*.

- ▶ Or specify a color (American spelling):

`It \textcolor{red}{stops}` `and \textcolor{green}{starts}.`	It stops and starts.

Presentations with `beamer`: Figures

- Use `\includegraphics` from the `graphicx` package.
- The `figure` environment centers by default, in `beamer`.

```
\begin{figure}
\includegraphics[
  width=0.5\textwidth]{gerbil}
\end{figure}
```


Image license: CC0

Presentations with `beamer`: Tables

- ▶ Tables in LATEX take some getting used to.
- ▶ Use the `tabular` environment from the `tabularx` package.
- ▶ The argument specifies column alignment — **l**eft, **r**ight, **r**ight.

```
\begin{tabular}{lrr}
Item  & Qty & Unit \$ \\
Widget & 1  & 199.99  \\
Gadget & 2  & 399.99  \\
Cable  & 3  & 19.99   \\
\end{tabular}
```

Item	Qty	Unit $
Widget	1	199.99
Gadget	2	399.99
Cable	3	19.99

- ▶ It also specifies vertical lines; use `\hline` for horizontal lines.

```
\begin{tabular}{|l|r|r|} \hline
Item  & Qty & Unit \$ \\\hline
Widget & 1  & 199.99  \\
Gadget & 2  & 399.99  \\
Cable  & 3  & 19.99   \\\hline
\end{tabular}
```

Item	Qty	Unit $
Widget	1	199.99
Gadget	2	399.99
Cable	3	19.99

- ▶ Use an ampersand ⟨&⟩ to separate columns and a double backslash ⟨\⟩⟨\⟩ to start a new row.

Presentations with `beamer`: Blocks

- A `block` environment makes a titled box.

  ```
  \begin{block}{Interesting Fact}
  This is important.
  \end{block}

  \begin{alertblock}{Cautionary Tale}
  This is really important!
  \end{alertblock}
  ```

  Interesting Fact

  This is important.

  Cautionary Tale

  This is really important!

- How exactly they look depends on the theme...

Presentations with `beamer`: Themes

- ▶ Customise the look of your presentation using themes.

```
\documentclass{beamer}

% or Warsaw, Bergen, Madrid, ...
\usetheme{Darmstadt}

% or albatross, beaver, crane, ...
\usecolortheme{beetle}

\title{Theme Demo}
\author{John}
\begin{document}
\begin{frame}
\titlepage
\end{frame}
\end{document}
```

- A frame can generate multiple slides.
- Use the `\pause` command to show only part of a slide.

```
\begin{itemize}
\item Can you feel the
\pause \item anticipation?
\end{itemize}
```

- Can you feel the

- A frame can generate multiple slides.
- Use the `\pause` command to show only part of a slide.

```
\begin{itemize}
\item Can you feel the
\pause \item anticipation?
\end{itemize}
```

- Can you feel the

- anticipation?

- There many more clever ways of making animations in beamer; see also the `\only`, `\alt`, and `\uncover` commands.

Drawings with Ti*k*Z

- ► Ti*k*Z is a package for drawing figures in LATEX.
- ► It defines a powerful drawing language inside LATEX. Short programs can draw surprisingly complicated things.
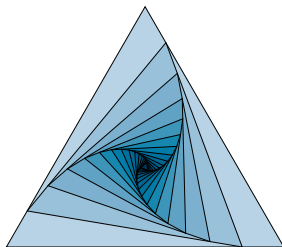
- ► We'll start with simple things. To draw a line in Ti*k*Z:

```
\begin{tikzpicture}
\draw (0,0) -- (1,1); % a line
\end{tikzpicture}
```

Drawings with TikZ: Coordinates

- The default coordinates are centimeters, with the usual sense:

$(0,3)$ $(3,3)$

$(0,0)$ $(3,0)$
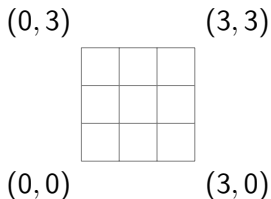
- It helps to draw a grid when you are working with TikZ:

```
\begin{tikzpicture}
\draw[help lines] (0,0) grid (3,3);
\end{tikzpicture}
```
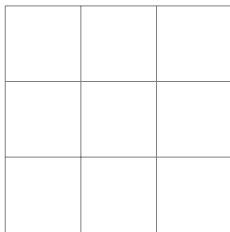
Drawings with TikZ: Lines

- Arrow heads and line styles are specified as options to the `\draw` command.
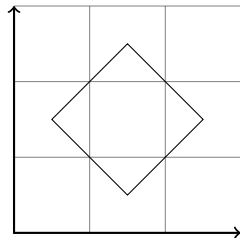- End each draw command with a `;` semicolon.

```
\begin{tikzpicture}
\draw[help lines] (0,0) grid (3,3);
\draw[->] (0,0) -- (1,1);
\draw[<->, thick] (2,1) -- (1,2);
\draw[<-, thick, dashed] (2,2)--(3,3);
\end{tikzpicture}
```
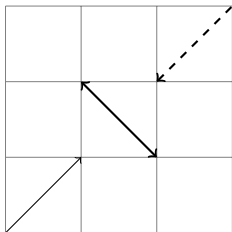
Drawings with TikZ: Paths

- You can specify multiple points to form a path.
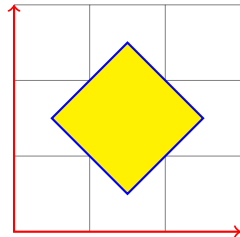- Arrows will appear only at the ends of the path.

```
\begin{tikzpicture}
\draw[help lines] (0,0) grid (3,3);
% axes:
\draw[<->, thick] (0,3)--(0,0)--(3,0);
% diamond:
\draw (1.5,0.5) -- (2.5,1.5) --
      (1.5,2.5) -- (0.5,1.5) --
      cycle; % close the path
\end{tikzpicture}
```

Drawings with TikZ: Colours
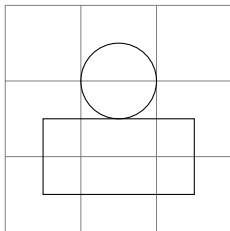
- Colours are also specified as options to `\draw`.

```
\begin{tikzpicture}
\draw[help lines] (0,0) grid (3,3);
% axes
\draw[<->, thick, red]
  (0,3)--(0,0)--(3,0);
% diamond
\draw[thick, blue, fill=yellow]
  (1.5,0.5) -- (2.5,1.5) --
  (1.5,2.5) -- (0.5,1.5) --
  cycle;
\end{tikzpicture}
```

Drawings with TikZ: Shapes
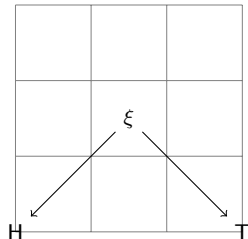
- ► TikZ has built-in commands for simple shapes.

```
\begin{tikzpicture}
\draw[help lines] (0,0) grid (3,3);
\draw (1.5,2.0) circle (0.5);
\draw (0.5,0.5) rectangle (2.5,1.5);
\end{tikzpicture}
```

Drawings with TikZ: Nodes & Labels

- Use nodes to place text (and math) in TikZ drawings.
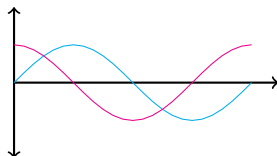- You can also use nodes as coordinates — useful for diagrams.

```
\begin{tikzpicture}
\draw[help lines] (0,0) grid (3,3);
\node (h) at (0,0) {H};
\node (x) at (1.5,1.5) {$\xi$};
\node (t) at (3,0) {T};
\draw[->] (x) -- (h);
\draw[->] (x) -- (t);
\end{tikzpicture}
```

Drawings with TikZ: Functions

▶ You can even plot some simple functions.

```
\begin{tikzpicture}[scale=0.5]
% y axis
\draw[<->, thick] (0,2) -- (0,-2);
% x axis
\draw[ ->, thick] (0,0) -- (7, 0);
% curves
\draw[cyan,domain=0:2*pi]
  plot (\x, {sin(\x r)});
\draw[magenta,domain=0:2*pi]
  plot (\x, {cos(\x r)});
\end{tikzpicture}
```

Drawings with TikZ: Examples

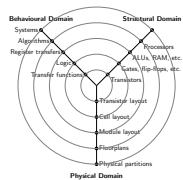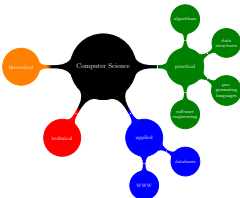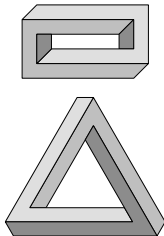- Check out TEXample.net for many TikZ examples:

Figure 1: Gajski-Kuhn Y-chart

Notes with `todonotes`

- The `\todo` command from the `todonotes` package is great for leaving notes to yourself and your collaborators.

  ```
  \todo{add results}
  \todo[color=blue!20]{fix method}
  ```
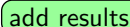
  > add results

  > fix method

- Pro Tip: define your own commands with `\newcommand`

  ```
  \newcommand{\alice}[1]{\todo[color=green!40]{#1}}
  \newcommand{\bob}[1]{\todo[color=purple!40]{#1}}
  ```

  This can save a lot of typing:

  ```
  \alice{add results}
  \bob{fix method}
  ```

  > add results

  > fix method

Notes with `todonotes`

- Only inline notes are supported with beamer, but margin notes are supported for normal documents.
- There is also a handy `\listoftodos` command.

Towards the Confusing Unification of Rasterization
and Local-Area Networks in State Machines

Alice Bob, Carol David, Edward Fredrick

Todo list

Abstract

Rasterization and Smalltalk, while important in theory, have not until recently been considered important. Given the current status of wearable methodologies, analysts clearly desire the refinement of IPv4. Purr, our new heuristic for the producer-consumer problem [1], is the solution to all of these problems.

1 Introduction

Recent advances in certifiable symmetries and Bayesian technology synchronize in order to realize access points. This is a direct result of the construction of multicast algorithms. This is a direct result of the analysis of active networks. The emulation of suffix trees would profoundly improve congestion control [4].

To our knowledge, our work in our research marks the first method analyzed specifically for scalable models.Existing interactive and permutable methodologies use Smalltalk to measure the construction of the partition table. The disadvantage of this type of method, however, is that hash tables can be made real-time, cooperative, and reliable. Existing "fuzzy" and concurrent algorithms use the evaluation of multicast frameworks to request access points. On the other hand, distributed archetypes might not be the

Are they polynomial time?

Realize multicast access points?

Instead of controlling the forward-error correction?

Phasellus libero ipsum, pellentesque sit amet, sem.

1

Spreadsheets with `spreadtab`

- ▶ Now that you've seen how LaTeX can replace Word and PowerPoint, what about Excel?
- ▶ Homework: try the `spreadtab package` !

Thanks, and happy T_EXing!