# A presentation in LaTeX beamer on TeX/LaTeX.

Jack Rosenthal

CSM Linux Users Group

24 September 2015

# TEX/LATEX

Jack Rosenthal

CSM Linux Users Group

24 September 2015

# lugtex.tex

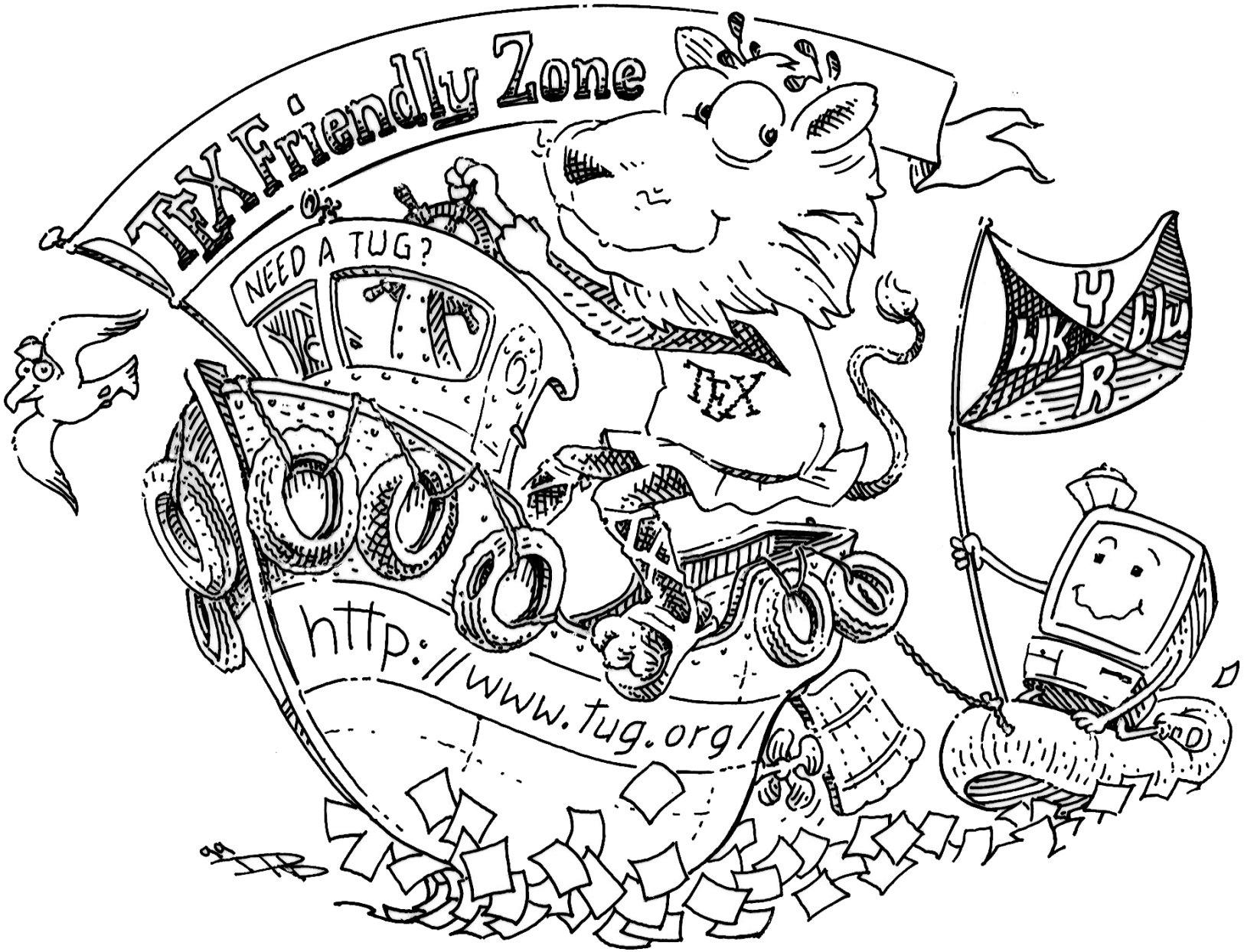
Jack Rosenthal

CSM Linux Users Group

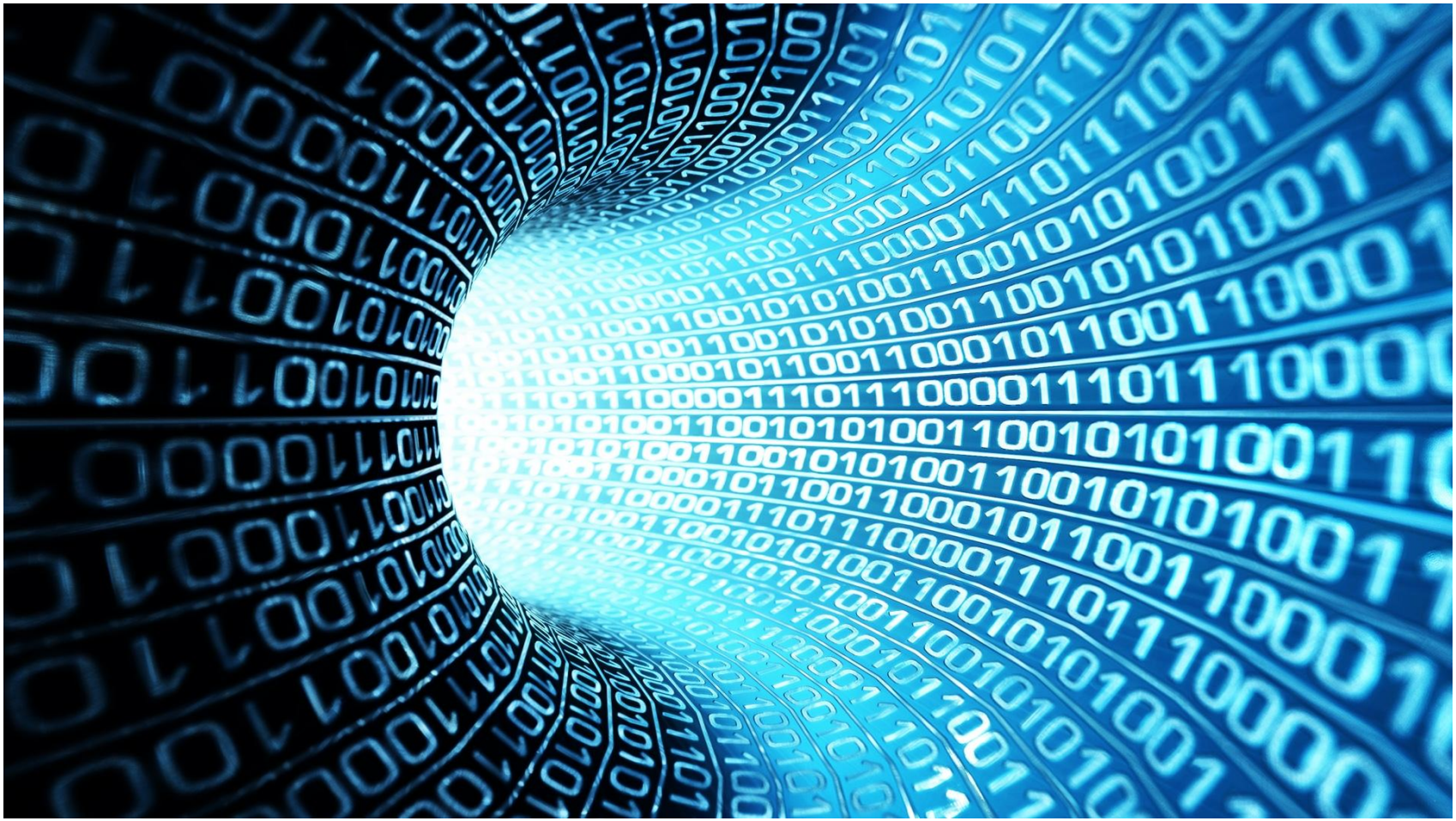
24 September 2015

# What are these things you say?

# What are these things you say?



# What are these things you say?



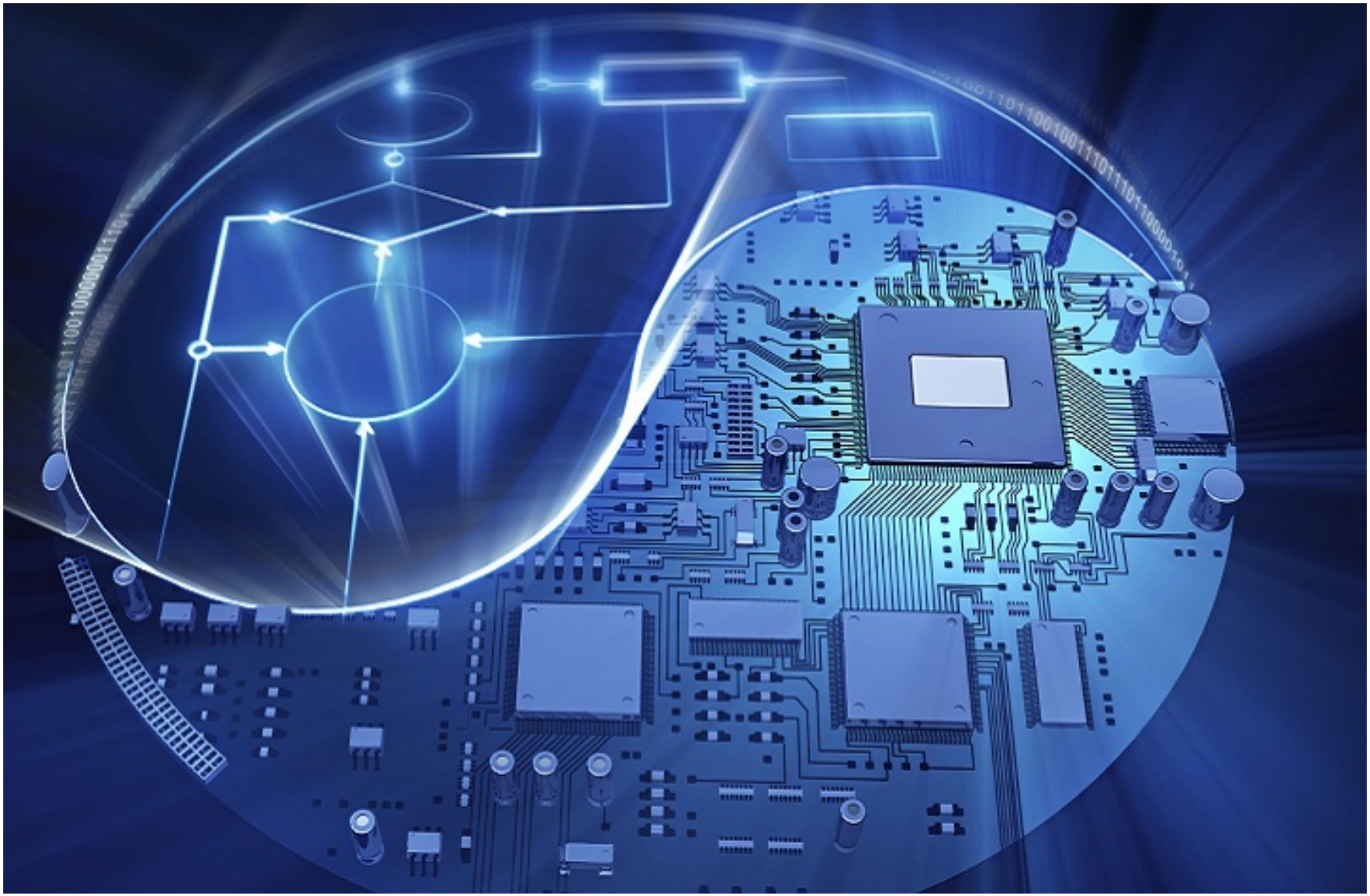
# What are these things you say?

## Typewriter Keyboard On Typesetting Machine

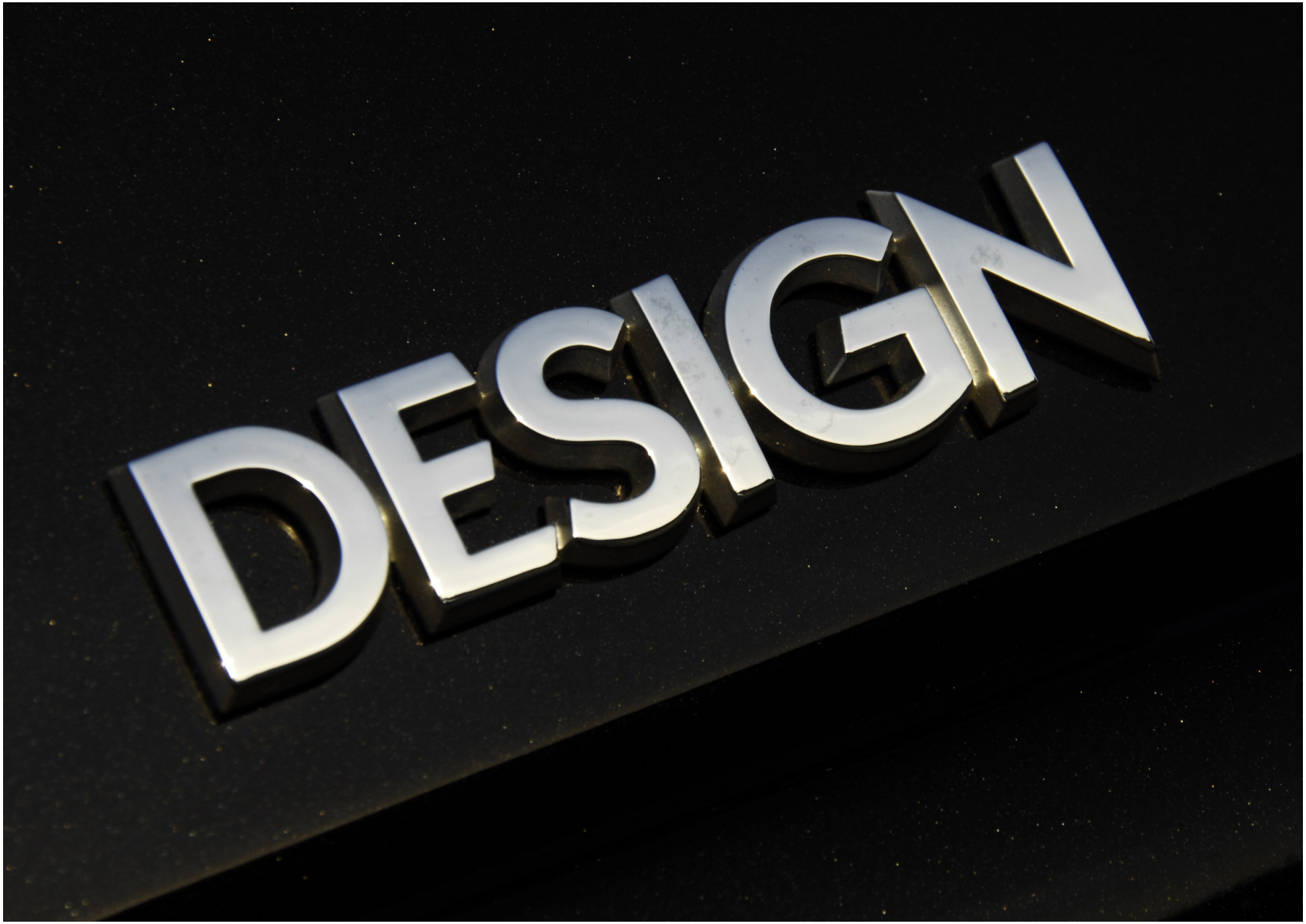
Typists can now set type on Linotype and Intertype composing machines through the development of a keyboard that has the standard keys of a typewriter. Forty-four keys, electrically operated, fit over the 90 keys of a standard composing machine. The keyboard can be moved from one composing machine to another as there is no installation. The new keyboard is simply placed over the top of the existing keyboard and the unit is ready for use when it is plugged into an electrical outlet. The complete outfit weighs only 25½ pounds.

Information not listed on articles in the index, starting on page 10, frequently is listed in the WHERE-TO-FIND-IT INDEX, which is available without charge from Bureau of Information, Popular Mechanics Magazine, 200 East Ontario Street, Chicago 11, Illinois

# What are these things you say?

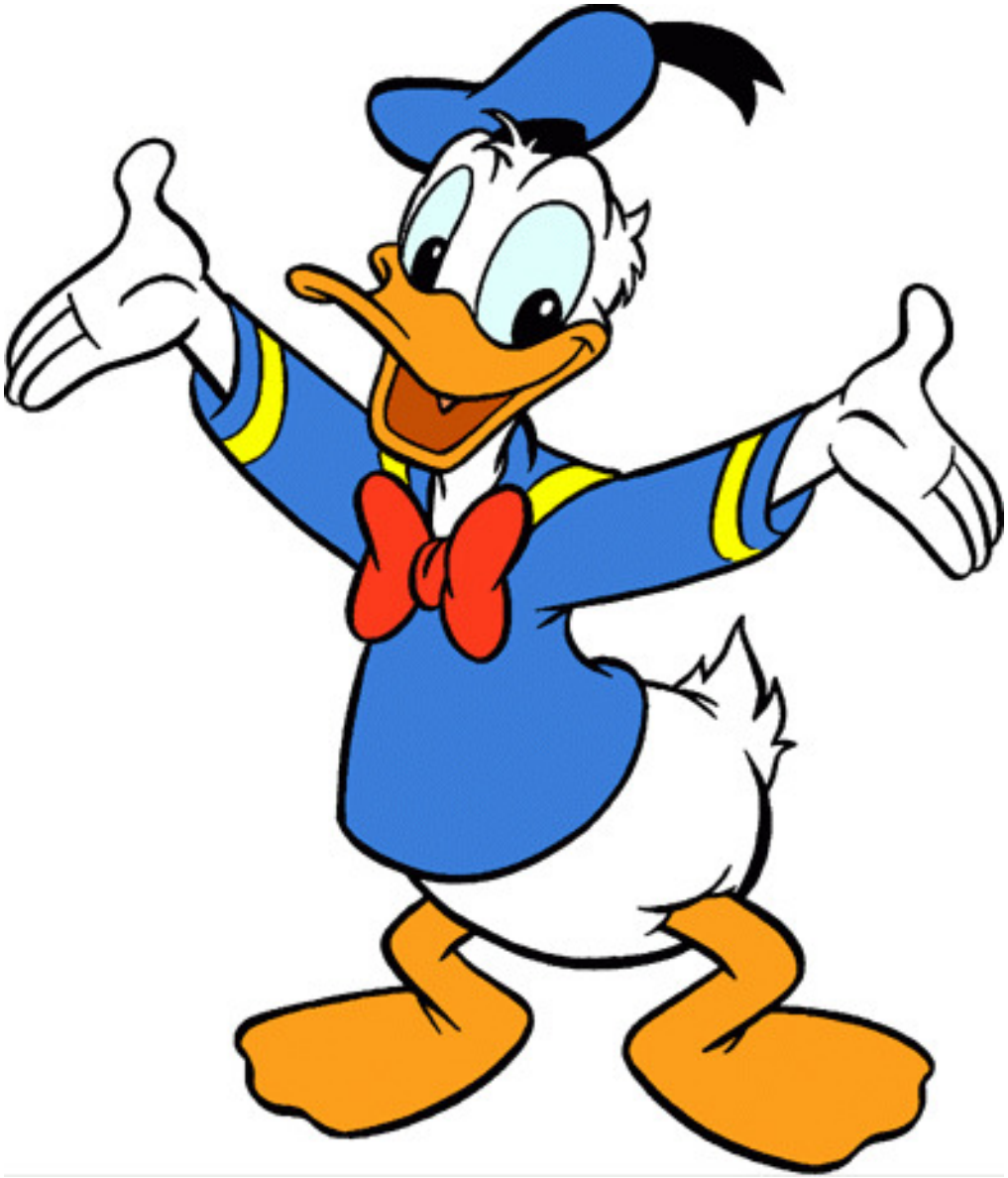


# What are these things you say?

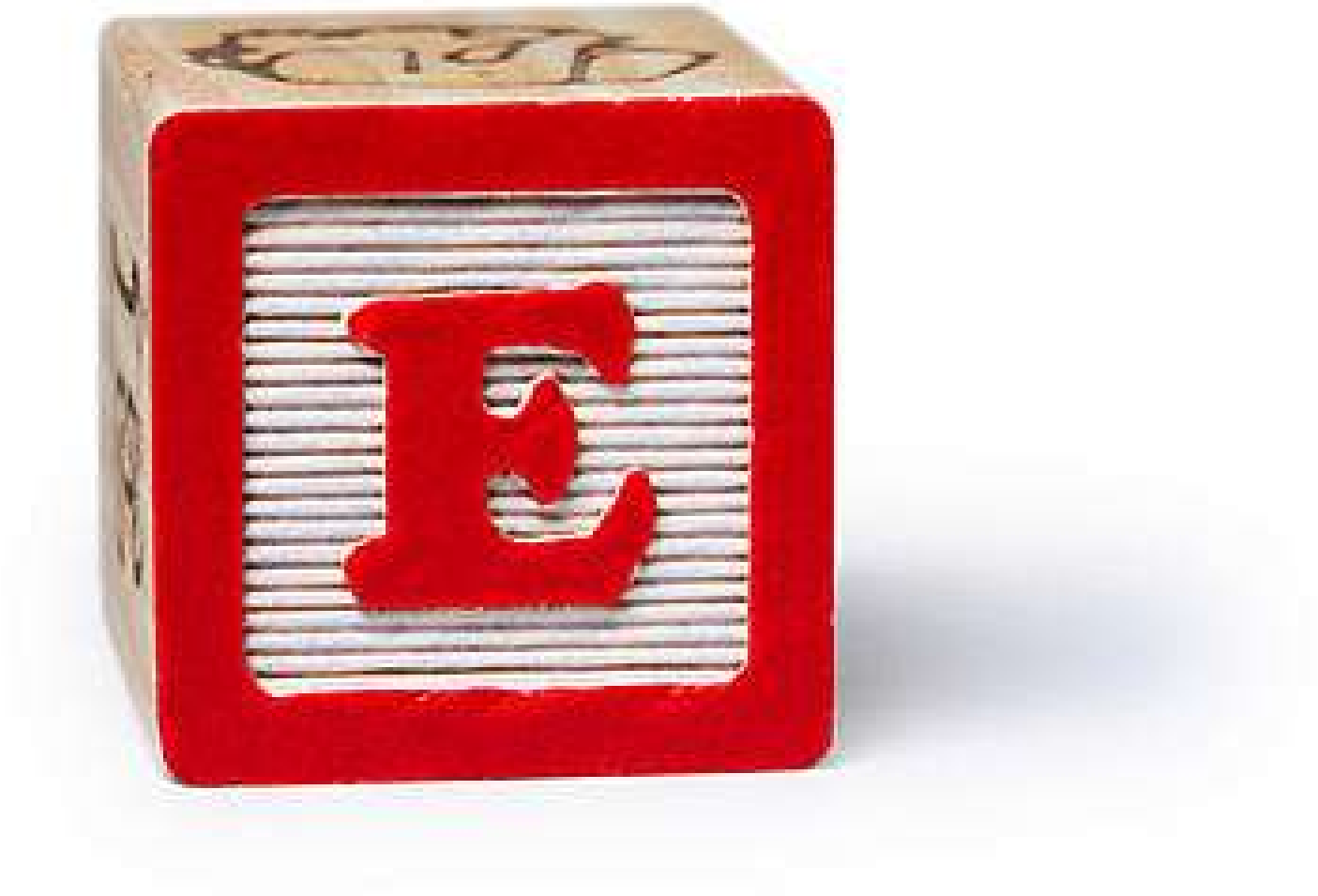


# What are these things you say?

# What are these things you say?



# What are these things you say?



# What are these things you say?

# What are these things you say?

```
for (;;)
```

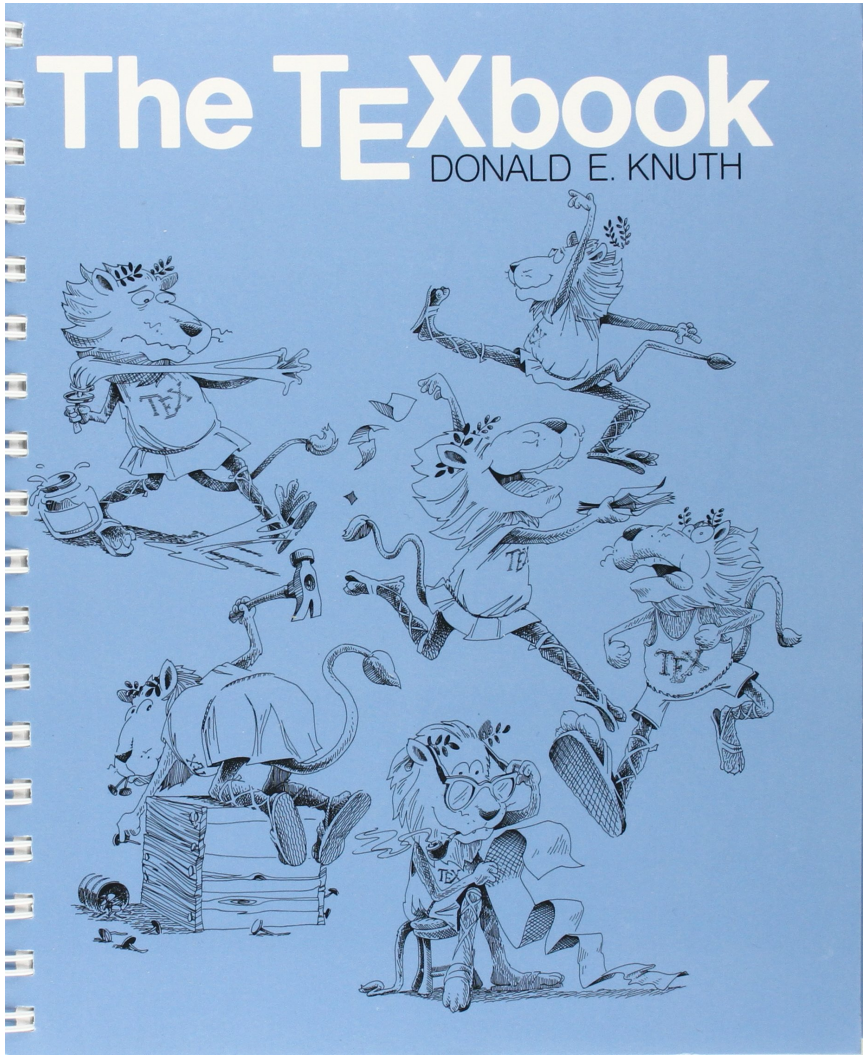
# What are these things you say?

# What are these things you say?

# What are these things you say?

# What are these things you say?

# What are these things you say?



# What are these things you say?

# tri: The Tokenize, Run, and Iterate Interpreter

Jack Rosenthal, CSCI-261 Section C

15 April 2015

## 1 Problem Description

The tri interpreter is an interpreter for a programming language I thought of called tri: tokenize, run, and iterate. The language is quite simple and does not include anything complex.

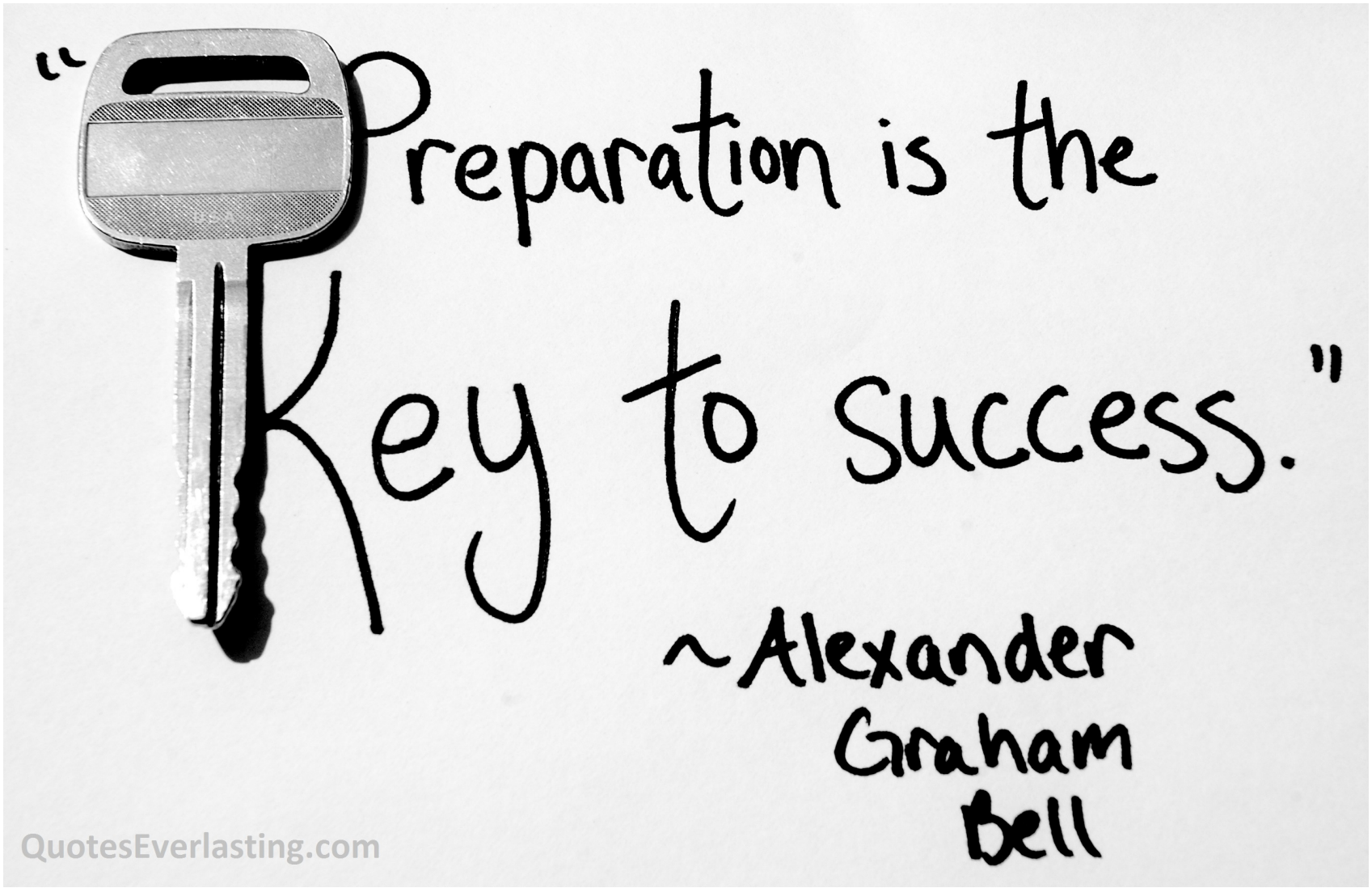
Unlike many languages that execute the source code at run time, the tri language provides type safety, allowing the programmer to create variables with a specific type like C and C++. Additionally, tri provides scoping, making variables declared in specific scopes only available in the specific scopes and removed from memory when the scope ends.

An example of some tri code is provided below:

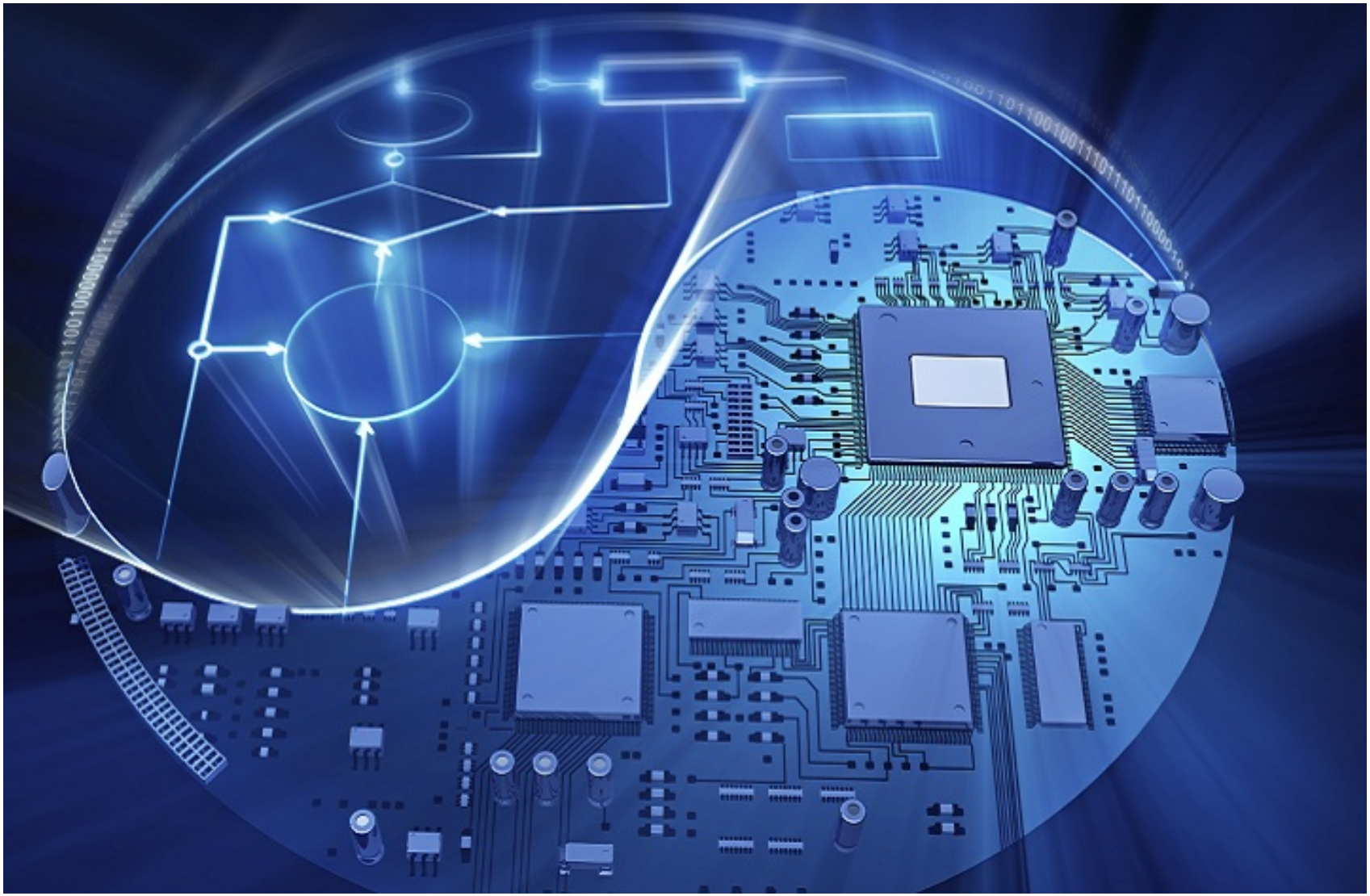
```
#!/usr/bin/tri
my char c;
c = 'q';
printf("%c", c); # tri supports C like printf notation
if (c == 'q') { # a new scope has been created
    my int32 someNumber;
    someNumber = 5;
    printf("%d", someNumber);
}

printf("%d", someNumber); #this would error
```

# What are these things you say?

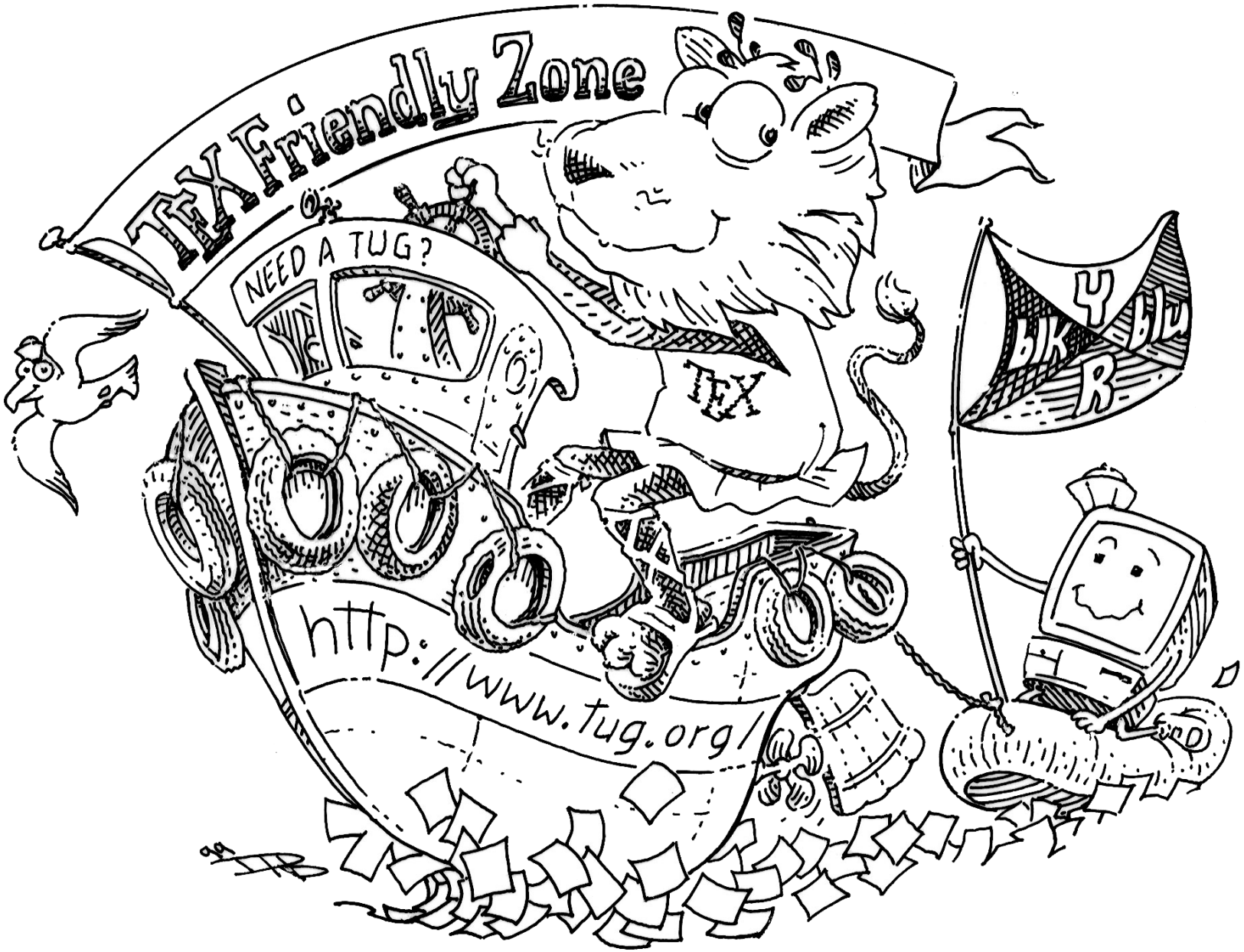


# What are these things you say?



# What are these things you say?

# What are these things you say?



# The Name of the Game

## From the TEXbook

English words like ‘technology’ stem from a Greek root beginning with the letters $\tau\epsilon\chi\ldots$; and this same Greek word means *art* as well as technology...

Insiders pronounce the $\chi$ of TEX as a Greek chi, not as an ‘x’, so that TEX rhymes with the word blecchhh... When you say it correctly to your computer, the terminal may become slightly moist.

# The Name of the Game

## From LaTeX: A Document Preparation System

One of the hardest things about LaTeX is deciding how to pronounce it. This is also one of the few things I'm not going to tell you about LaTeX, since pronunciation is best determined by usage, not fiat. TeX is usually pronounced teck, making lah-teck, and lay-teck the logical choices.

# Typesetting vs. Ordinary Typing

- Typesetting is the art of putting letters on a page

# Typesetting vs. Ordinary Typing

- Typesetting is the art of putting letters on a page
- Ligatures appear in professional typesetting, such as in the word find (rather than find)

# Typesetting vs. Ordinary Typing

- Typesetting is the art of putting letters on a page
- Ligatures appear in professional typesetting, such as in the word find (rather than find)
- Typesetting can involve complex mathematics, of which TEX handles quite well

$$\sum_{n=0}^{\infty} a_n z^n \quad \text{converges if} \quad |z| < \left(\limsup_{n\to\infty} \sqrt[n]{|a_n|}\right)^{-1}.$$

# Trying to Math in Word

**LaTeX**

$$-\int_0^{2\pi} \frac{kQ\,d\theta}{2\pi(a^2+x^2)^{3/2}}(a\sin\theta\,\hat{j}) = 0 \tag{1}$$

**Word**

$$-\int_0^{2\pi} \frac{kQd\theta}{2\pi\left(a^2+x^2\right)^{3/2}}\left(a\sin\theta\ \hat{j}\right) = 0$$

# Structure of a LaTeX Document

Preamble

```
\documentclass[12pt]{article}
\usepackage[margin=25mm]{geometry}
```

```
\begin{document}
```

Body

```
Hello World from \LaTeX !
\begin{equation}
  \sum_{n = 0}^\infty {x^n \over n!} = e^x
\end{equation}
```

```
\end{document}
```

# Structure of a LaTeX Document

Preamble
```
\documentclass[12pt]{article}
\usepackage[margin=25mm]{geometry}
```
```
\begin{document}
```
Body
```
Hello World from \LaTeX !
\begin{equation}
  \sum_{n = 0}^\infty {x^n \over n!} = e^x
\end{equation}
```
```
\end{document}
```

## Output

Hello World from LaTeX!

$$\sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x \tag{2}$$

# Control Sequences

- Immediately after typing ‘\’, TEX expects a control word or symbol.

# Control Sequences

- Immediately after typing '\', TEX expects a control word or symbol.
- A control word constists of a backslash followed by one or more *letters*.

# Control Sequences

- Immediately after typing '\', TEX expects a control word or symbol.
- A control word constists of a backslash followed by one or more *letters*.
- A control symbol constists of a backslash followed by a single *nonletter*.

# Control Sequences

- Immediately after typing '\', TEX expects a control word or symbol.
- A control word constists of a backslash followed by one or more *letters*.
- A control symbol constists of a backslash followed by a single *nonletter*.
- Example: '`\input MS`' causes TEX to begin reading a file called '`MS.tex`'.

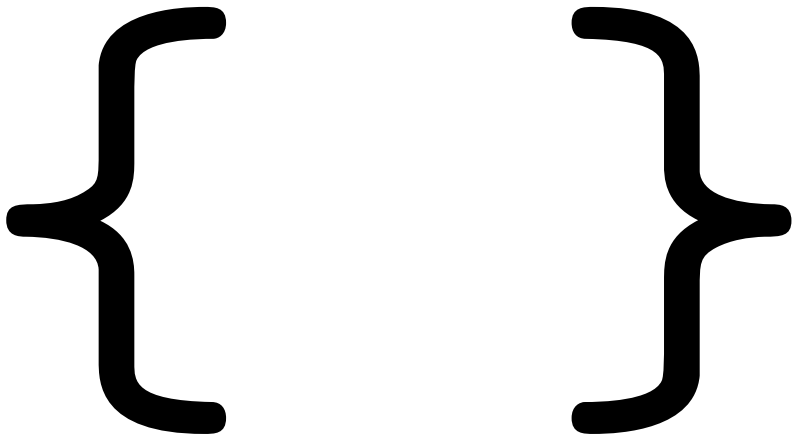
# Control Sequences

- Immediately after typing '\', TEX expects a control word or symbol.
- A control word constists of a backslash followed by one or more *letters*.
- A control symbol constists of a backslash followed by a single *nonletter*.
- Example: '`\input MS`' causes TEX to begin reading a file called '`MS.tex`'.
- Example: TEX converts '`George P\'olya and Gabor Szeg\"o`' to 'George Pólya and Gabor Szegö.'

# Control Sequences

- Immediately after typing '\', TEX expects a control word or symbol.
- A control word constists of a backslash followed by one or more *letters*.
- A control symbol constists of a backslash followed by a single *nonletter*.
- Example: '`\input MS`' causes TEX to begin reading a file called '`MS.tex`'.
- Example: TEX converts
  '`George P\'olya and Gabor Szeg\"o`' to 'George Pólya and Gabor Szegö.'
- A space after a control word is ignored, to fix this, escape the space afted a control word when required.
  `\TeX\ ignores spaces after control words.`

# Grouping



# Fonts of Type in LaTeX

`\textrm{This}` This produces roman typeface output.
`\textsl{This}` *This* produces slanted typeface output.
`\textit{This}` *This* produces italics typeface output.
`\textbf{This}` **This** produces bold typeface output.
`\texttt{This}` `This` produces typewriter typeface output.
`\textsf{This}` This produces sans typeface output.

# Installing TEX Live

Arch Linux:

```
# pacman -S texlive-most
```

Debian/Ubuntu/Mint:

```
# apt-get install texlive-full
```

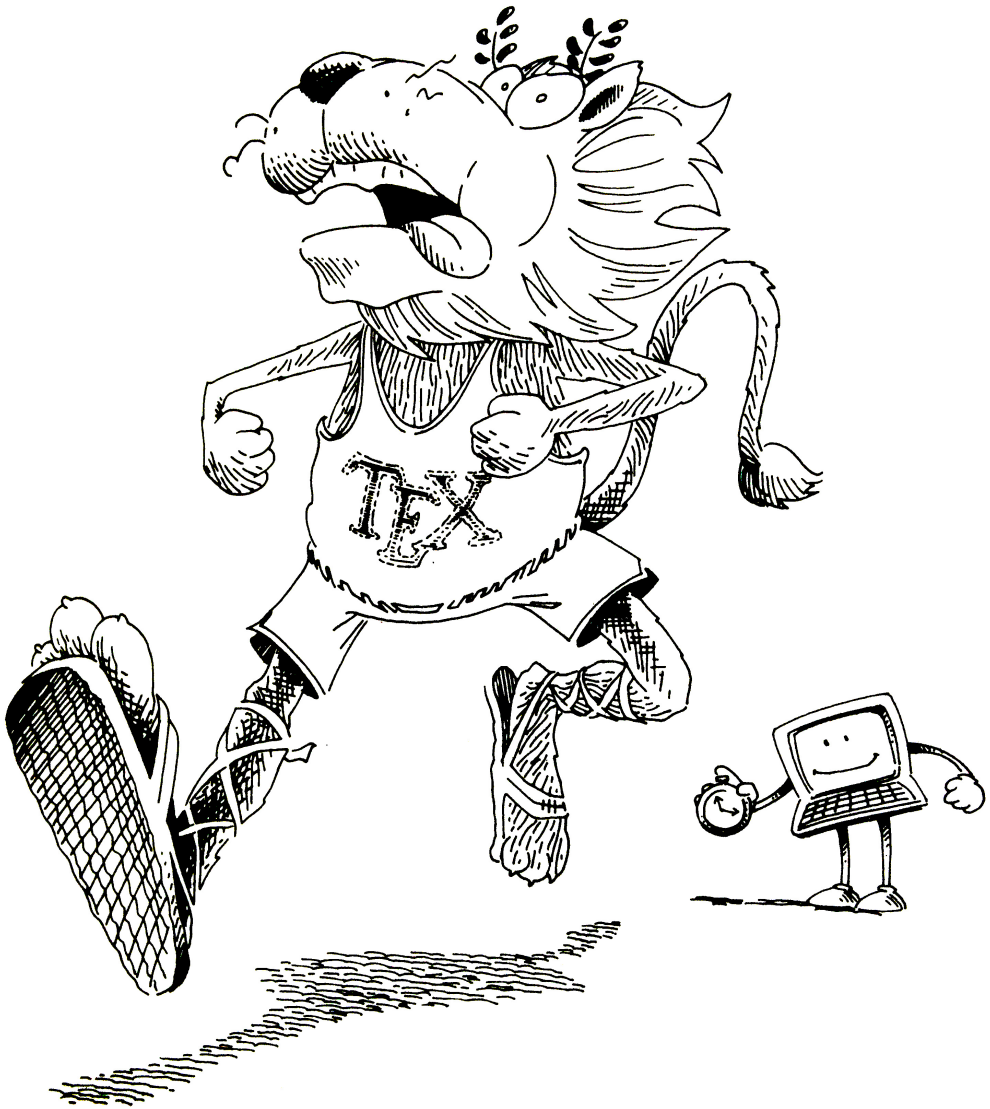
Fedora:

```
# yum install texlive
```

Windows/OS X:

Follow instructions at `http://tug.org`

# Running TEX



# Running LaTeX

When you start `pdflatex`, you will see the following:

```
This is pdfTeX, Version 3.14159265-2.6-1.40.16
(TeX Live 2015) (preloaded format=pdflatex)
**
```

# Running LaTeX

When you start `pdflatex`, you will see the following:

```
This is pdfTeX, Version 3.14159265-2.6-1.40.16
(TeX Live 2015) (preloaded format=pdflatex)
**
```

The '**' is TeX's way of asking you for an input filename. If you don't want to type in the filename through standard input each time, provide the filename as the first argument.

# Running LaTeX

When you start `pdflatex`, you will see the following:

```
This is pdfTeX, Version 3.14159265-2.6-1.40.16
(TeX Live 2015) (preloaded format=pdflatex)
**
```

The '`**`' is TeX's way of asking you for an input filename. If you don't want to type in the filename through standard input each time, provide the filename as the first argument.

To use TeX in a REPL like manner, type '`\relax`' at the prompt for a filename. This is TeX's NoOp command, in this case you are using it to tell TeX to expect nothing after an '`\input`'.

# Running LaTeX

When you start `pdflatex`, you will see the following:

```
This is pdfTeX, Version 3.14159265-2.6-1.40.16
(TeX Live 2015) (preloaded format=pdflatex)
**
```

The '`**`' is TeX's way of asking you for an input filename. If you don't want to type in the filename through standard input each time, provide the filename as the first argument.

To use TeX in a REPL like manner, type '`\relax`' at the prompt for a filename. This is TeX's NoOp command, in this case you are using it to tell TeX to expect nothing after an '`\input`'.

pdfTeX will produce a PDF file.

# The documentclass

When you start a document, you start it with a line that reads something like this:

```
\documentclass{article}
```

# The documentclass

When you start a document, you start it with a line that reads something like this:

```
\documentclass{article}
```

There are actually many documentclasses to choose from:

- article
- IEEEtran
- proc
- minimal
- report
- book
- memior
- letter
- exam
- leaflet
- beamer
- ...

# The documentclass

When you start a document, you start it with a line that reads something like this:

```
\documentclass{article}
```

There are actually many documentclasses to choose from:

- article
- IEEEtran
- proc
- minimal
- report
- book
- memior
- letter
- exam
- leaflet
- beamer
- ...

You can also specify options like this:

```
\documentclass[12pt,a4paper,titlepage]{article}
```

# Environments

environment

```
...
\begin{itemize}
    \item An item
    \item Another item
\end{itemize}
...
```

# Environments

environment

```
...
\begin{itemize}
    \item An item
    \item Another item
\end{itemize}
...
```

## Output

- An item
- Another item

# Environments

environment

```
...
\begin{enumerate}
    \item An item
    \item Another item
\end{enumerate}
...
```

## Output

1. An item
2. Another item

# Mathematics in LaTeX

Use the `equation` environment for basic equation displays:

```
\begin{equation}
    \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{
        \Delta x}
\end{equation}
```

# Mathematics in LaTeX

Use the `equation` environment for basic equation displays:

```
\begin{equation}
    \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{
        \Delta x}
\end{equation}
```

## Output

$$\lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{3}$$

# Mathematics in LaTeX

Use the `equation` environment for basic equation displays:

```
\begin{equation}
    \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{
        \Delta x}
\end{equation}
```

## Output

$$\lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{3}$$

Or use '`$ ... $`' to quickly show math in a paragraph:

```
... we can see that as $x \to \infty$, the amount of ...
```

# Mathematics in LaTeX

Use the `equation` environment for basic equation displays:

```
\begin{equation}
    \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{
        \Delta x}
\end{equation}
```

## Output

$$\lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \tag{3}$$

Or use '`$ ... $`' to quickly show math in a paragraph:

```
... we can see that as $x \to \infty$, the amount of ...
```

## Output

... we can see that as $x \to \infty$, the amount of ...

# How TEX Reads What You Type

- A ⟨return⟩ is like a space

# How TEX Reads What You Type

- A ⟨return⟩ is like a space
- Two spaces in a row count as a single space

# How TEX Reads What You Type

- A ⟨return⟩ is like a space
- Two spaces in a row count as a single space
- A blank line denotes the end of a paragraph

# How TEX Reads What You Type

- A ⟨return⟩ is like a space
- Two spaces in a row count as a single space
- A blank line denotes the end of a paragraph

TEX also categorgizes your characters. There are 16 categories as follows:

| *Cat* | *Meaning* | *Default* | *Cat* | *Meaning* | *Default* |
|---|---|---|---|---|---|
| 0 | Escape character | `\` | 8 | Subscript | `_` |
| 1 | Begin group | `{` | 9 | Ignored characer | ⟨null⟩ |
| 2 | End group | `}` | 10 | Space | ⟨space⟩ |
| 3 | Math shift | `$` | 11 | Letter | `A-Z,a-z` |
| 4 | Alignment tab | `&` | 12 | Other character | |
| 5 | End of line | ⟨return⟩ | 13 | Active character | `~` |
| 6 | Parameter | `#` | 14 | Comment character | `%` |
| 7 | Superscript | `^` | 15 | Invalid character | ⟨delete⟩ |

Don't worry too much about this. All this means is that you will have to escape a few special characters.

# Line Breaking

- TEX has a badness value from 0 to 10,000, where 0 is perfect and 10,000 is infinitely bad, for almost everything in your document that is flexible.

# Line Breaking

- TEX has a badness value from 0 to 10,000, where 0 is perfect and 10,000 is infinitely bad, for almost everything in your document that is flexible.
- When a line is perfect in spacing between words and no hyphenation, the badness will be zero.

# Line Breaking

- TEX has a badness value from 0 to 10,000, where 0 is perfect and 10,000 is infinitely bad, for almost everything in your document that is flexible.
- When a line is perfect in spacing between words and no hyphenation, the badness will be zero.
- As words get too tight or too narrow, the badness increases.

# Line Breaking

- TEX has a badness value from 0 to 10,000, where 0 is perfect and 10,000 is infinitely bad, for almost everything in your document that is flexible.
- When a line is perfect in spacing between words and no hyphenation, the badness will be zero.
- As words get too tight or too narrow, the badness increases.
- Hyphenation in words adds a lot of badness!

# Line Breaking

- TEX has a badness value from 0 to 10,000, where 0 is perfect and 10,000 is infinitely bad, for almost everything in your document that is flexible.
- When a line is perfect in spacing between words and no hyphenation, the badness will be zero.
- As words get too tight or too narrow, the badness increases.
- Hyphenation in words adds a lot of badness!
- TEX then optimises the badness of each line, trying to get it as low as possible.

# Sectioning

Use the commands:

```
\part{Part I} % only in the book class
\chapter{Awesome Chapter} % only in book, report
\section{Optimal Awesome}
\subsection{Here It Is}
```

# Sectioning

Use the commands:

```
\part{Part I} % only in the book class
\chapter{Awesome Chapter} % only in book, report
\section{Optimal Awesome}
\subsection{Here It Is}
```

Then you can generate your table of contents using:

```
\tableofcontents
```

# Graphics

In your preamble, include the `graphicx` package:

```
\usepackage{graphicx}
```

Then in your body: (arguments and file extension optional)

```
\includegraphics[width=4cm]{coolpix.png}
```

# Tables

‘&’ acts as an alignment character in the `tabular` environment, ‘\\’ acts as a newline:

```
\begin{tabular}{ |l|l| }
    \hline
    stuff & stuff \\ \hline
    stuff & stuff \\
    \hline
\end{tabular}
```

# Tables

'&' acts as an alignment character in the `tabular` environment, '\\' acts as a newline:

```
\begin{tabular}{ |l|l| }
    \hline
    stuff & stuff \\ \hline
    stuff & stuff \\
    \hline
\end{tabular}
```

## Output

| | |
|---|---|
| stuff | stuff |
| stuff | stuff |

# Tables

'&' acts as an alignment character in the `tabular` environment, '\\' acts as a newline:

```
\begin{tabular}{ |l|l| }
    \hline
    stuff & stuff \\ \hline
    stuff & stuff \\
    \hline
\end{tabular}
```

## Output

| | |
|---|---|
| stuff | stuff |
| stuff | stuff |

Also take a look at the excellent `tabu` package.

# Automagically Numbered Floating Figures and Tables

```
\begin{figure}
    \centering
    \includegraphics[width=4cm]{coolpix}
    \caption{This is a really cool picture}
\end{figure}
```

# Automagically Numbered Floating Figures and Tables

```
\begin{figure}
    \centering
    \includegraphics[width=4cm]{coolpix}
    \caption{This is a really cool picture}
\end{figure}

\begin{table}
    \centering
    \caption{Important Data About Stuff}
    \begin{tabular}{ | l | l | c | }
        ...
    \end{tabular}
\end{table}
```

# Automagically Numbered Floating Figures and Tables

```
\begin{figure}
    \centering
    \includegraphics[width=4cm]{coolpix}
    \caption{This is a really cool picture}
\end{figure}

\begin{table}
    \centering
    \caption{Important Data About Stuff}
    \begin{tabular}{ | l | l | c | }
        ...
    \end{tabular}
\end{table}
```

You can also generate a List of Figures and a List of Tables:

```
\listoffigures
\listoftables
```

# Presentations

Use the `beamer` class, slides are in the `frame` environment.

```
\documentclass{beamer}
\begin{document}
\begin{frame}
    \frametitle{Relevant Title}
    Hello World!
    \pause % Advance slide to continue
    This won't show till you click.
    \begin{block}{Cool Information}
        This shows in a fancy blue block
    \end{block}
\end{frame}
\end{document}
```

# Relevant Title

Hello World!

# Relevant Title

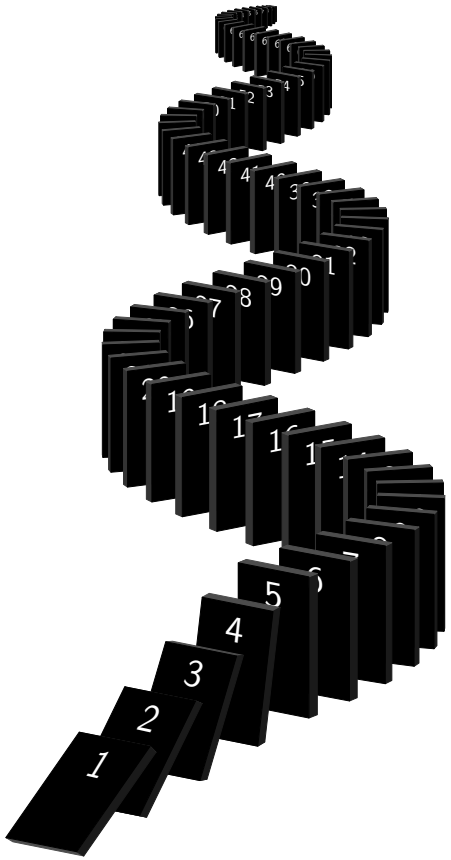
Hello World! This won't show till you click.

**Cool Information**

This shows in a fancy blue block

# Cool Tricks

Using Ti*k*Z...



# Comprehensive TEX Archive Network (CTAN)

CTAN
is a great site that has all of the
TEX and LATEX packages and sources.
`http://ctan.org`

# Resources and Recommended Reading

- The TEXbook, Donald E. Knuth

# Resources and Recommended Reading

- The TEXbook, Donald E. Knuth
- LATEX: A Document Preparation System, Leslie Lamport

# Resources and Recommended Reading

- The TEXbook, Donald E. Knuth
- LATEX: A Document Preparation System, Leslie Lamport
- The LATEX Companion, Frank Mittelbach & Michel Goossens

# Resources and Recommended Reading

- The TEXbook, Donald E. Knuth
- LATEX: A Document Preparation System, Leslie Lamport
- The LATEX Companion, Frank Mittelbach & Michel Goossens
- The LATEX Book on Wikibooks

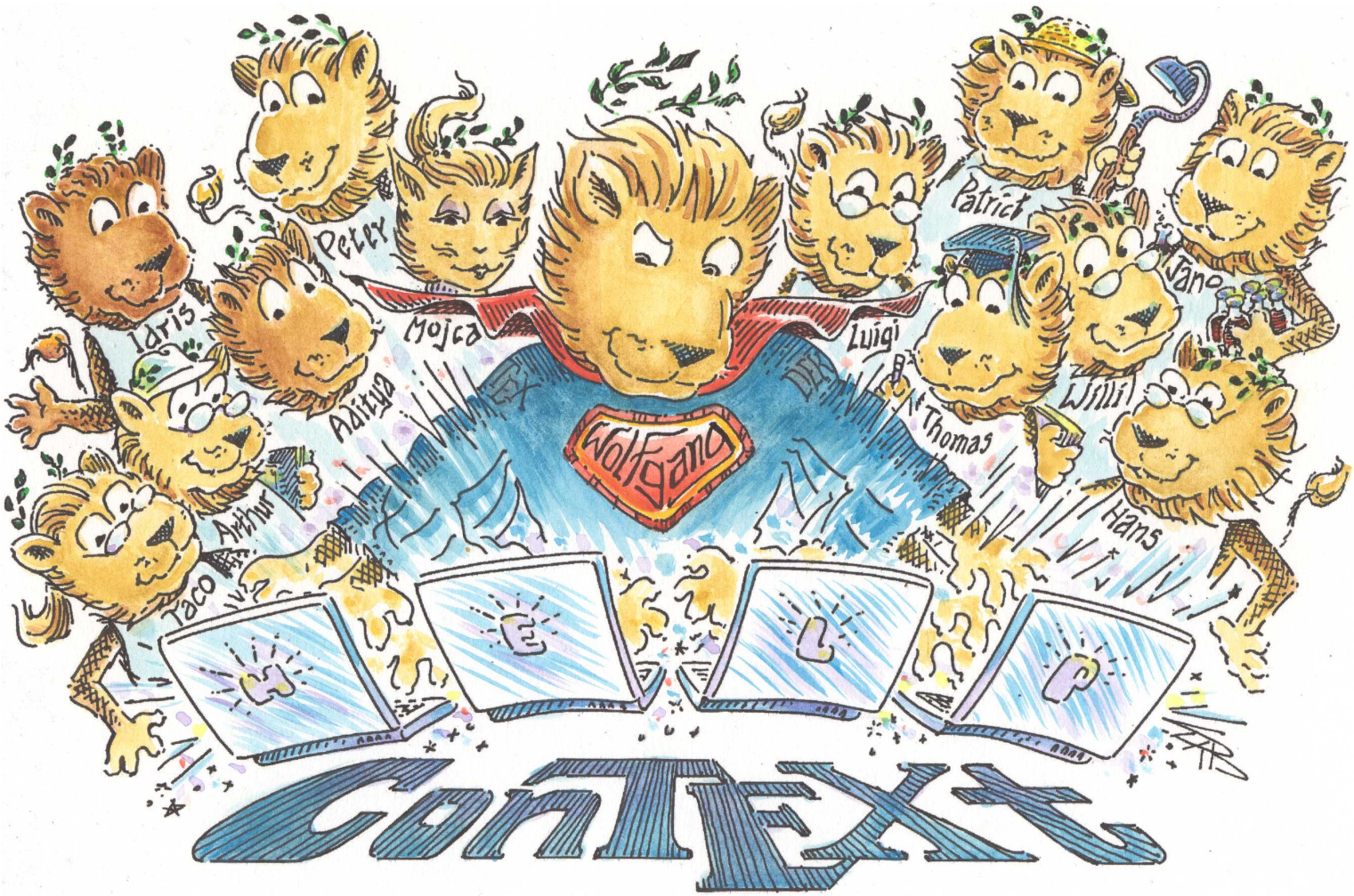
# Resources and Recommended Reading

- The TEXbook, Donald E. Knuth
- LATEX: A Document Preparation System, Leslie Lamport
- The LATEX Companion, Frank Mittelbach & Michel Goossens
- The LATEX Book on Wikibooks
- `http://texdoc.net`

# Resources and Recommended Reading

- The TEXbook, Donald E. Knuth
- LATEX: A Document Preparation System, Leslie Lamport
- The LATEX Companion, Frank Mittelbach & Michel Goossens
- The LATEX Book on Wikibooks
- `http://texdoc.net`
- TEX Users Group: `http://tug.org`

# Getting Help



# Getting Help

You likely haven't found a bug in TEX. Knuth pays `0x$80.00` for every bug found in the current stable versions of TEX and METAFONT.

- TEX Stack Exchange
- `texhax` mailing list
- Come to LUG!