LaTeX Workshop

David Branner

Hacker School, New York City
April 10, 2013

1 Introduction

2 Math typesetting

3 Some tools of use in documenting code

4 Styling non-math text

5 Slides

6 Automating the production of PDFs as program output

7 Turing-completeness and its consequences

8 References

9 The Lore of TeX

What LaTeX is

LaTeX is a high-level implementation of TeX.

What LaTeX is

LaTeX is a high-level implementation of TeX.

TeX is basically a math typesetting-system. It has been generalized to be useful in application to a variety of typographic and word-processing tasks.

What LaTeX is

LaTeX is a high-level implementation of TeX.

TeX is basically a math typesetting-system. It has been generalized to be useful in application to a variety of typographic and word-processing tasks.

Beyond that, it is also superbly useful for automatically generating high-quality PDFs from the output of a computer program.

TeX operates mainly through markup and formatting commands, though the tagging looks quite different from more modern systems such as HTML and Markdown.

TeX operates mainly through markup and formatting commands, though the tagging looks quite different from more modern systems such as HTML and Markdown.

(TeX dates from 1978, putting it in the same generation of long-lived tools as vi, EMACS, awk, and sed — not to mention C and SQL.)

Important ideas:

Important ideas:

1. **symbols**: `abcDEF`, `123`, `.;!`, etc.

Important ideas:

1. **symbols**: `abcDEF`, `123`, `.;!`, etc.
2. **values**: `10`, `10pt`, used in settings and calculation

Important ideas:

1. **symbols**: `abcDEF`, `123`, `.;!`, etc.
2. **values**: `10`, `10pt`, used in settings and calculation
3. **commands**: `\somecommand[options]{argument}`

Important ideas:

1. **symbols**: `abcDEF`, `123`, `.;!`, etc.
2. **values**: `10`, `10pt`, used in settings and calculation
3. **commands**: `\somecommand[options]{argument}`
4. **environments**

Important ideas:

1. **symbols**: `abcDEF`, `123`, `.;!`, etc.
2. **values**: `10`, `10pt`, used in settings and calculation
3. **commands**: `\somecommand[options]{argument}`
4. **environments**:

```
\begin{someenvironment}
...
\end{someenvironment}
```

Math typesetting

Math typesetting is the first skill you should learn, since there are tools to make this easy.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}

See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}
`~`	"nobreak space"	`a~b` vs. `a b`	$a~b$ vs. ab

See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}
`~`	"nobreak space"	`a~b` vs. `a b`	$a\ b$ vs. ab
`\delta`	δ	`\Delta`	Δ

See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}
`~`	“nobreak space”	`a~b` vs. `a b`	$a\ b$ vs. ab
`\delta`	δ	`\Delta`	Δ
`\sin x`	$\sin x$	cf. `sin x`	$sinx$

See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}
`~`	"nobreak space"	`a~b` vs. `a b`	$a~b$ vs. ab
`\delta`	δ	`\Delta`	Δ
`\sin x`	$\sin x$	cf. `sin x`	$sinx$
`\ldots`	$\ldots$	cf. `...~\cdots`	$...~\cdots$

See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}
`~`	“nobreak space”	`a~b` vs. `a b`	$a~b$ vs. ab
`\delta`	δ	`\Delta`	Δ
`\sin x`	$\sin x$	cf. `sin x`	$sinx$
`\ldots`	$\ldots$	cf. `...~\cdots`	... $\cdots$
`\int`	$\int$	`\int_a^b`	$\int_a^b$
`\sum`	$\sum$	`\sum_{n=0}^\infty`	$\sum_{n=0}^\infty$

See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}
`~`	"nobreak space"	`a~b` vs. `a b`	$a\ b$ vs. ab
`\delta`	δ	`\Delta`	Δ
`\sin x`	$\sin x$	cf. `sin x`	$sinx$
`\ldots`	$\ldots$	cf. `...~\cdots`	$...~\cdots$
`\int`	$\int$	`\int_a^b`	$\int_a^b$
`\sum`	$\sum$	`\sum_{n=0}^\infty`	$\sum_{n=0}^\infty$
`\not`	negation strike-through	`\in, \not\in`	$\in, \not\in$

See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}
`~`	“nobreak space”	`a~b` vs. `a b`	$a~b$ vs. $a b$
`\delta`	δ	`\Delta`	Δ
`\sin x`	$\sin x$	cf. `sin x`	$sin x$
`\ldots`	$\ldots$	cf. `...~\cdots`	... $\cdots$
`\int`	$\int$	`\int_a^b`	$\int_a^b$
`\sum`	$\sum$	`\sum_{n=0}^\infty`	$\sum_{n=0}^\infty$
`\not`	negation strike-through	`\in, \not\in`	$\in, \not\in$
`\vec{}`	vector diacritic	`\vec{a}`	$\vec{a}$

See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
`^`	superscript	`a^{b+c}`	a^{b+c}
`_`	subscript	`a_{b+c}`	a_{b+c}
`~`	“nobreak space”	`a~b` vs. `a b`	$a~b$ vs. $a b$
`\delta`	δ	`\Delta`	Δ
`\sin x`	$\sin x$	cf. `sin x`	$sin x$
`\ldots`	$\ldots$	cf. `...~\cdots`	$...~\cdots$
`\int`	$\int$	`\int_a^b`	$\int_a^b$
`\sum`	$\sum$	`\sum_{n=0}^\infty`	$\sum_{n=0}^\infty$
`\not`	negation strike-through	`\in, \not\in`	$\in, \not\in$
`\vec{}`	vector diacritic	`\vec{a}`	$\vec{a}$
`\mathbb{}`	math “blackboard” font	`\mathbb{R}`	$\mathbb{R}$

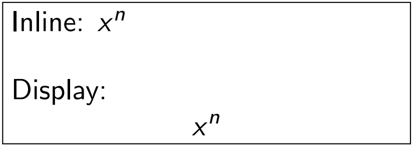
See slide #26 for information about hands-on practice.

Table : Important types of commands and operators in math

code	meaning	example or comparison	output
^	superscript	a^{b+c}	a^{b+c}
_	subscript	a_{b+c}	a_{b+c}
~	"nobreak space"	a~b vs. a b	$a\ b$ vs. ab
\delta	δ	\Delta	Δ
\sin x	$\sin x$	cf. sin x	$sinx$
\ldots	$\ldots$	cf. ...~\cdots	$\ldots\ \cdots$
\int	$\int$	\int_a^b	$\int_a^b$
\sum	$\sum$	\sum_{n=0}^\infty	$\sum_{n=0}^\infty$
\not	negation strike-through	\in, \not\in	$\in, \notin$
\vec{}	vector diacritic	\vec{a}	$\vec{a}$
\mathbb{}	math "blackboard" font	\mathbb{R}	$\mathbb{R}$
\frac{}{}	fraction	\{x}{y}	$\frac{x}{y}$

See slide #26 for information about hands-on practice.

“Inline” vs. “display” math environments



Inline: x^n

Display:

$$x^n$$

```
Inline: \(x^n\)
\vskip12pt
Display: \[x^n\]
```

Inline: x_n

Display:

$$x_n$$

```
Inline: \(x_n\)
\vskip12pt
Display: \[x_n\]
```

$$\boxed{\lim_{n\rightarrow\infty}\frac{1}{n}=0}$$

```
\[\lim_{n\rightarrow\
    infty}
 \frac{1}{n}=0\]
```

$$\sum_{n=1}^{\infty}\frac{1}{n^2}=\frac{\pi^2}{6}$$

```
\[\sum_{n=1}^{\infty}
 \frac{1}{n^2}=
 \frac{\pi^2}{6}\]
```

$$\boxed{\int_a^b x\ dx = \frac{b^2-a^2}{2}}$$

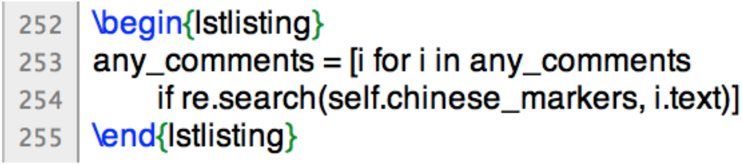
```
\[\int_a^b x\ dx =
 \frac{b^2-a^2}{2}\]
```

Listing program code

Output:

```
any_comments = [i for i in any_comments
          if re.search(self.chinese_markers, i.text)]
```

Code that produced the output above:



```
\begin{lstlisting}
any_comments = [i for i in any_comments
          if re.search(self.chinese_markers, i.text)]
\end{lstlisting}
```

Tables and graphics

We'll use a real-time example of `table` or `tabular`. (leaving the slides)

Aligned formulas

We'll use a real-time example of `align`. (leaving the slides)

Basic text styling

Basic styling of text for **boldface**, *italics*, SMALL CAPS, underlining, ~~strikethrough~~ (the last two using the `ulem` package), etc.

```
Basic styling of text for
     \textbf{boldface}, \
    textit{italics}, \
    textsc{small caps}, \
    uline{underlining}, \
    sout{strikethrough} (
    the last two using
    the \texttt{ulem}
    package), etc.
```

There are countless others. Notice that these operate transparently via a kind of markup.

Footnotes and cross-references

Footnotes are placed into text using `\footnote{}`:

Footnotes and cross-references

Footnotes are placed into text using `\footnote{}`:

A label can be placed in the text as `\label{someLabelName}` and then referred to as `\ref{someLabelName}`. You'll see some examples below, on slide #20.

Footnotes and cross-references

Footnotes are placed into text using `\footnote{}`:

A label can be placed in the text as `\label{someLabelName}` and then referred to as `\ref{someLabelName}`. You'll see some examples below, on slide #20.

Considerably more elaborate behaviors are available through specialized packages.

Text filler

Here is paragraph number six of the standard text-fill.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

```
\usepackage{lipsum}
\lipsum[6]
```

Inserting graphics

The following code inserts a graphic up to the maximum width of the text-area available.

```
\usepackage{calc}     % allows calculations
\newlength{\imgwidth} % declares ‘variable’
\newcommand\grw[1]{   % declares ‘function’
 \settowidth{\imgwidth}{\includegraphics{#1}}%
 \setlength{\imgwidth}
   {\minof{\imgwidth}{\textwidth}}%
 \includegraphics[width=\imgwidth]{#1}%
}
```

Getting help

The slide above is probably a little frightening.

Getting help

The slide above is probably a little frightening.

The number of packages and lower-level commands you are likely to need is not actually very large for most purposes. But it is daunting for most people at first, so I recommend using a manual (Mittelbach *et al.* is the best — see slide #30).

Getting help

The slide above is probably a little frightening.

The number of packages and lower-level commands you are likely to need is not actually very large for most purposes. But it is daunting for most people at first, so I recommend using a manual (Mittelbach *et al.* is the best — see slide #30).

I also recommend using on-line resources (discussed on slide 27).

Getting help

The slide above is probably a little frightening.

The number of packages and lower-level commands you are likely to need is not actually very large for most purposes. But it is daunting for most people at first, so I recommend using a manual (Mittelbach *et al.* is the best — see slide #30).

I also recommend using on-line resources (discussed on slide 27).

The command above appears in the preamble. Let me illustrate what the document body of a real-world example looks like. (leaving the slides)

Slides

We'll use a real-time example of `beamer`. (leaving the slides)

Automating the production of PDFs as program output

We'll use a real-time example from a program of mine. (leaving the slides)

Automating the production of PDFs as program output

We'll use a real-time example from a program of mine. (leaving the slides)

Briefly, we place the preamble and any static pieces of LaTeX code into files whose contents can be read in, and then we either use templating or other string-services to typeset the data.

Turing-completeness and its consequences

TeX is a Turing-complete language.

As such, it is capable of some surprising effects:

As such, it is capable of some surprising effects:

As such, it is capable of some surprising effects:

1. Coffee stains

As such, it is capable of some surprising effects:

1. Coffee stains: the `coffee4` package

As such, it is capable of some surprising effects:

1. Coffee stains: the `coffee4` package
2. A BASIC interpreter

As such, it is capable of some surprising effects:

1. Coffee stains: the `coffee4` package
2. A BASIC interpreter
3. `sudokubundle`: A Sudoku puzzle generator, solver, and template

As such, it is capable of some surprising effects:

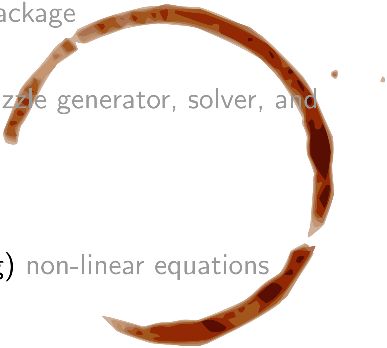
1. **Coffee stains**: the `coffee4` package
2. A BASIC interpreter
3. `sudokubundle`: A Sudoku puzzle generator, solver, and template
4. Controller for a Mars Rover

As such, it is capable of some surprising effects:

1. Coffee stains: the `coffee4` package
2. A BASIC interpreter
3. `sudokubundle`: A Sudoku puzzle generator, solver, and template
4. Controller for a Mars Rover
5. Reversi game (“reverxii”)

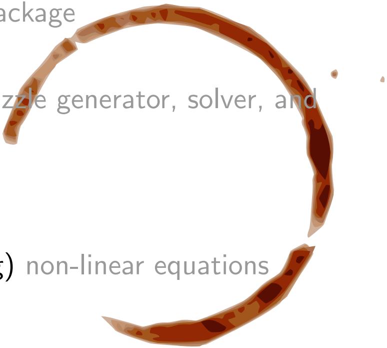
As such, it is capable of some surprising effects:

1. Coffee stains: the `coffee4` package
2. A BASIC interpreter
3. `sudokubundle`: A Sudoku puzzle generator, solver, and template
4. Controller for a Mars Rover
5. Reversi game ("reverxii")
6. Solving (not merely displaying) non-linear equations



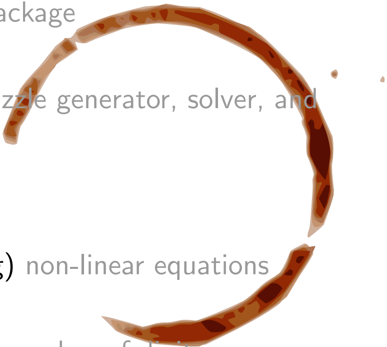
As such, it is capable of some surprising effects:

1. **Coffee stains**: the `coffee4` package
2. A BASIC interpreter
3. `sudokubundle`: A Sudoku puzzle generator, solver, and template
4. Controller for a Mars Rover
5. Reversi game ("reverxii")
6. Solving (not merely displaying) non-linear equations
7. Game of Life



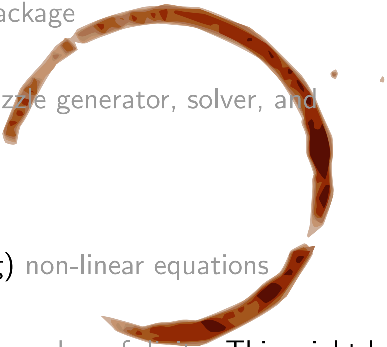
As such, it is capable of some surprising effects:

1. Coffee stains: the `coffee4` package
2. A BASIC interpreter
3. `sudokubundle`: A Sudoku puzzle generator, solver, and template
4. Controller for a Mars Rover
5. Reversi game (“reverxii”)
6. Solving (not merely displaying) non-linear equations
7. Game of Life
8. Calculating pi to a user-input number of digits



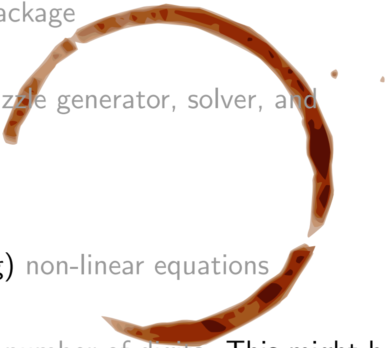
As such, it is capable of some surprising effects:

1. Coffee stains: the `coffee4` package
2. A BASIC interpreter
3. `sudokubundle`: A Sudoku puzzle generator, solver, and template
4. Controller for a Mars Rover
5. Reversi game ("reverxii")
6. Solving (not merely displaying) non-linear equations
7. Game of Life
8. Calculating pi to a user-input number of digits. This might be worth looking at. One important thing to notice is that it prints to the console — even though PDF is what we normally think of as our "output", producing it is actually a secondary step after the running of a TeX program.



As such, it is capable of some surprising effects:

1. Coffee stains: the `coffee4` package
2. A BASIC interpreter
3. `sudokubundle`: A Sudoku puzzle generator, solver, and template
4. Controller for a Mars Rover
5. Reversi game ("reverxii")
6. Solving (not merely displaying) non-linear equations
7. Game of Life
8. Calculating pi to a user-input number of digits. This might be worth looking at. One important thing to notice is that it prints to the console — even though PDF is what we normally think of as our "output", producing it is actually a secondary step after the running of a TeX program.
9. Lambda calculus



The list seems endless. . .

1. How do I make my document look like it was written by a Cthulhu-worshipping madman? — especially the graphic below the answer here
2. Letterpress effect through PSTricks or Tikz
3. TikZ and PGF examples
4. One answer to "What is the most bizarre thing you have seen done with TeX?"
5. Self-replication
6. Text spirals

See `http://tex.stackexchange.com/questions/tagged/fun`.

To practice math syntax, or for graphical self-help, go to CodeCogs site and choose the “standalone editor.”

Applications of LaTeX syntax:

1. MathJax, for use in HTML
2. Description of use in MediaWiki
3. Go to WolframAlpha and enter

   ```
   \int_0^\infty\frac{x}{e^x}dx
   ```

   and then hit return. It should display (and then solve)

$$\int_0^\infty \frac{x}{e^x} dx$$

Best help sites:

Best help sites:

1. TeX-LaTeX Stack Exchange (A superbly supportive forum environment. You'll find many other questions answered capably elsewhere on the stackoverflow.com site, too.)

Best help sites:

1. TeX-LaTeX Stack Exchange (A superbly supportive forum environment. You'll find many other questions answered capably elsewhere on the stackoverflow.com site, too.)
2. TeX Users Group.

Best help sites:

1. TeX-LaTeX Stack Exchange (A superbly supportive forum environment. You'll find many other questions answered capably elsewhere on the stackoverflow.com site, too.)
2. TeX Users Group.
3. LaTeX Community.

The main site for packages is CTAN: Comprehensive TeX Archive Network (maintained by the TEX Users Group)

It would be hard to list all the on-line tutorials and reference documentation, but here are three:

It would be hard to list all the on-line tutorials and reference documentation, but here are three:

1. Wikibooks' LaTeX guide

It would be hard to list all the on-line tutorials and reference documentation, but here are three:

1. Wikibooks' LaTeX guide
2. Nicola Talbot, *LaTeX for Complete Novices*

It would be hard to list all the on-line tutorials and reference documentation, but here are three:

1. Wikibooks' LaTeX guide
2. Nicola Talbot, *LaTeX for Complete Novices*
3. Help with LaTeX

Don't forget about **paper reference works**:
These two items (also listed on http://www.tug.org/books/, among many others)

Don't forget about **paper reference works**:
These two items (also listed on `http://www.tug.org/books/`, among many others)

1. Frank Mittelbach, Goossens, *et al.*, *The LaTeX Companion*. (Frank Mittelbach tells me a very good PDF version is supposed to appear on the Pearson website shortly.)

Don't forget about **paper reference works**:
These two items (also listed on http://www.tug.org/books/, among many others)

1. Frank Mittelbach, Goossens, *et al.*, *The LaTeX Companion*. (Frank Mittelbach tells me a very good PDF version is supposed to appear on the Pearson website shortly.)
2. Donald E. Knuth, *The TeXbook*. (Source at http://www.ctan.org/pkg/texbook

Don't forget about **paper reference works**:
These two items (also listed on http://www.tug.org/books/, among many others)

1. Frank Mittelbach, Goossens, *et al.*, *The LaTeX Companion*. (Frank Mittelbach tells me a very good PDF version is supposed to appear on the Pearson website shortly.)
2. Donald E. Knuth, *The TeXbook*. (Source at http://www.ctan.org/pkg/texbook; note that it is licensed in such a way as to "prevent distribution" —please respect the author's wishes.

Don't forget about **paper reference works**:
These two items (also listed on `http://www.tug.org/books/`, among many others)

1. Frank Mittelbach, Goossens, *et al.*, *The LaTeX Companion*. (Frank Mittelbach tells me a very good PDF version is supposed to appear on the Pearson website shortly.)
2. Donald E. Knuth, *The TeXbook*. (Source at `http://www.ctan.org/pkg/texbook`; note that it is licensed in such a way as to "prevent distribution" —please respect the author's wishes. The `.tex` file is full of amusing commented remarks that do not appear in the PDF or printed volume.)

The Lore of TeX. . .

Version numbering

Version-numbering of TeX:

Version numbering

Version-numbering of TeX:
1

Version numbering

Version-numbering of TeX:
1
2

Version numbering

Version-numbering of TeX:

1

2

3

Version numbering

Version-numbering of TeX:
1
2
3.1

Version numbering

Version-numbering of TeX:
1
2
3.14

Version numbering

Version-numbering of TeX:
1
2
3.141

Version numbering

Version-numbering of TeX:
1
2
3.1415

Version numbering

Version-numbering of TeX:
1
2
3.14159

Version numbering

Version-numbering of TeX:
1
2
3.141592

Version numbering

Version-numbering of TeX:
1
2
3.1415926

Version numbering

Version-numbering of TeX:
1
2
3.1415926. That's all so far.

The pronunciation of TeX etc.

Donald Knuth:

> *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. Hence the name TeX, which is an uppercase form of $\tau\epsilon\chi$.*
>
> *Insiders pronounce the χ of TeX as a Greek chi, not as an 'x', so that TeX rhymes with the word blecchhh. It's the 'ch' sound in Scottish words like loch or German words like ach; it's a Spanish 'j' and a Russian 'kh'. When you say it correctly to your computer, the terminal may become slightly moist.*

The TeXbook, v. 3.0 (1996), p. 1

Separating content from formatting

Advocates of TeX often say that it enables you to separate content and formatting, allowing you to concentrate on the former.

Separating content from formatting

Advocates of TeX often say that it enables you to separate content and formatting, allowing you to concentrate on the former.

That would be true of any mark-up language, not just TeX.

Separating content from formatting

I have two reasons for doubting the value of this argument.

Separating content from formatting

I have two reasons for doubting the value of this argument.

First, in reality, doing a good job with TeX can take considerably longer than simply typing what you have to say into a standard word processor or text editor. A simpler mark-up language would be correspondingly more effective than TeX at saving time and fuss spent on formatting. Good TeX code is often quite complex.

Separating content from formatting

I have two reasons for doubting the value of this argument.

First, in reality, doing a good job with TeX can take considerably longer than simply typing what you have to say into a standard word processor or text editor. A simpler mark-up language would be correspondingly more effective than TeX at saving time and fuss spent on formatting. Good TeX code is often quite complex.

I don't think this objection is answerable.

Separating content from formatting, cont'd

Second: cognitively, human brains do not distinguish form and content very well. Think of someone shouting at you — do you really keep the message separate from how it is delivered?

Separating content from formatting, cont'd

Second: cognitively, human brains do not distinguish form and content very well. Think of someone shouting at you — do you really keep the message separate from how it is delivered?

I admit that that's an objection about reception of information. As a strategy for producing text, however, distinguishing the two may be useful.

Trivia about TeX

It was originally invented to enable mathematical typesetting. Complexity followed.

In the process of developing it, Knuth had to deal with the mathematics of both fonts and line-breaking, both highly non-trivial subjects.

About Donald Knuth

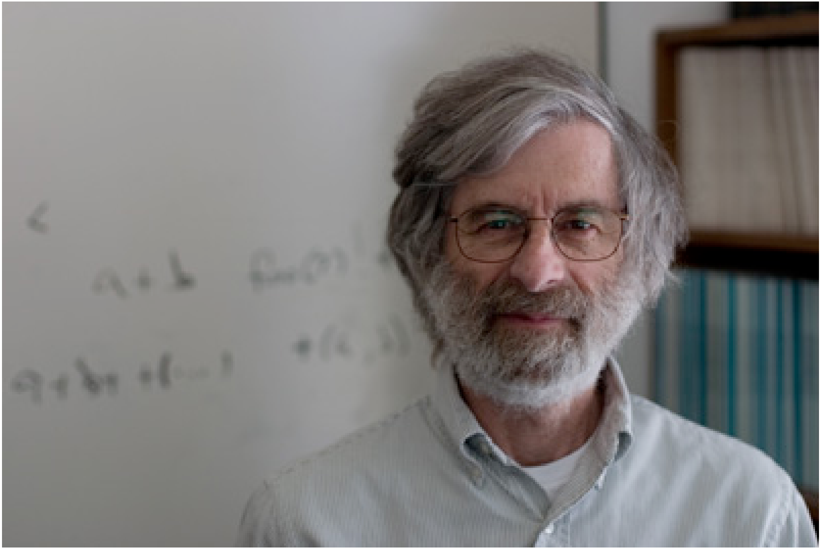
Famous for developing several important algorithms, including one for the fast matching of strings.

His books are permeated with humor. I am reminded of Bronowski's comment:

> *If you read Galileo's Dialogues and all those corny jokes and all that leg pulling, here is a man who is in love with his subject. . .*

Jacob Bronowski (1908–1974), *Magic, Science, and Civilization* (New York: Columbia University Press, 1978), p. 36.

Donald Knuth `http://www-cs-faculty.stanford.edu/~uno/dek-14May10-2.jpeg`, accessed 20130409. Taken by Dasha Slobozhanina at Case Western Reserve U.



Leslie Lamport http://www.budiu.info/blog/2007/05/03/an-interview-with-leslie-lamport/, accessed 20130409. Taken by Mihai Budiu.

END