Towards LaTeX Coding Standards

Didier Verna

didier@lrde.epita.fr
http://www.lrde.epita.fr/˜didier

TUG 2011 – Thursday, October 20

Table of contents

History

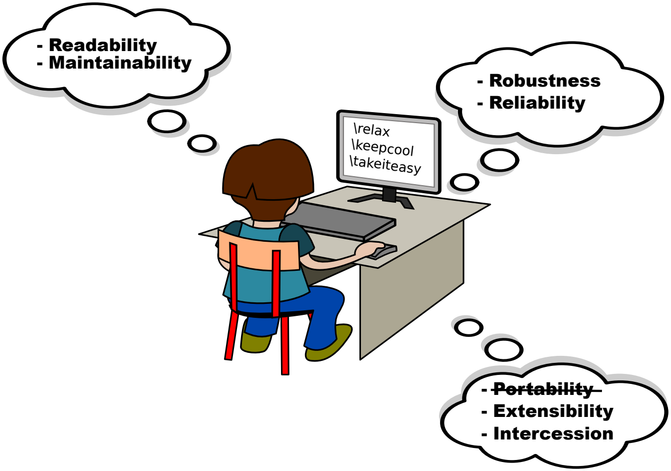
1918	*The Elements of Style*, William Strunk, Jr. and E.B. White (4th edition 1999). Style guide for writing American English.
1974	*The Elements of Programming Style*, B.W. Kernighan and P.J. Plauger, McGraw Hill (2nd Edition 1978). Style guide for programming.
...	*The Elements of* `whatnot` *Style*.

Purpose

- help programmers to read and understand source code
- not only their own but that of others
- From the GNU Coding Standards:

> *Their purpose is to make the GNU system clean, consistent, and easy to install. This document can also be read as a guide to writing portable, robust and reliable programs.*

The coding standards many-festos



- **Consistency:** the exact coding style is less important than actually sticking to it!

30 years and no style?

. . . makes LaTeX a dull toy. . .

- **Legacy**
  - Learning by example (learn the good *and* the bad)
- **Lack of help**
  - Liberal language (macro-expansion)
  - Editor support (complicated)
- **Lack of need**
  - A world of dwarfs
    (TEXLive 2009: average 327 LoC, median 134 LoC)
  - Antisocial development
    (most packages single-authored)

30 years and *almost* no style?
…makes LaTeX an *almost* dull toy…

- Tools
  - Blank lines, comment syntax
  - `calc`, `ifthen`, `doc`, `ltxdoc` ***etc.***
- Conventions
  - `\usepackage` ***vs.*** `\RequirePackage`, `@` character *etc.*
  - LaTeX itself not even conformant (*e.g.* `\hbox`, `\m@ne`)
- Companion
  - Section 2.1 (Structuring of source files)
  - Section A.4 (Structuring of package files)
  - Less than 1% of the book…

The need for coding standards is real

- **Why?**
  - Learning by *good* example
  - Facilitate interaction
  - Clean up the current intercession mess
- **How?**
  - Level 1 (low): layout – formatting, indentation, naming schemes *etc.*
  - Level 2 (mid): design – modularity, encapsulation, other paradigms *etc.*
  - Level 3 (high): behavior – user interfaces, package interaction / conflict management *etc.*
  - Level 4 (meta): social

Blanks

Be spacey

1. Stay WYSIWYGly coherent
   - `\\`, `\par`
   - Tabular-like environments (`&`, `\\`)
2. Put only one "logical" instruction per line
   - environment calls
   - `\expandafter\foo\bar`
   - `\raggedleft\foo\bar baz`
3. Be as spacey as you like in math mode
   - blanks ignored
4. Grouping (*any* kind) $\Longrightarrow$ indentation
   - `{}`, `\b[egin]group` `\e[nd]group`, `\makeatletter`, `\makeatother` *etc.*
5. The `%` character is your friend!

Example

Choose your preferred style…

```
\def\@docinclude#1 {\clearpage
\if@filesw \immediate\write\@mainaux{\string\@input{#1.aux}}\fi
\@tempswatrue\if@partsw \@tempswafalse\edef\@tempb{#1}\@for
\@tempa:=\@partlist\do{\ifx\@tempa\@tempb\@tempswatrue\fi}\fi
\if@tempswa \let\@auxout\@partaux \if@filesw
\immediate\openout\@partaux #1.aux
\immediate\write\@partaux{\relax}\fi
% ... \fi :-(
```

```
\def\@docinclude#1{
  \clearpage
  \if@filesw\immediate\write\@mainaux{\string\@input{#1.aux}}\fi
  \@tempswatrue
  \if@partsw
    \@tempswafalse
    \edef\@tempb{#1}
    \@for\@tempa:=\@partlist\do{\ifx\@tempa\@tempb\@tempswatrue\fi}
  \fi
  \if@tempswa
    \let\@auxout\@partaux
    \if@filesw
      \immediate\openout\@partaux #1.aux
      \immediate\write\@partaux{\relax}
    \fi
  % ... \fi :-)
```

Braces for impact!
Where do you put them?

Hmmm

```
\newenvironment{env}[1]
{%
   \opening\code
   \opening\code
}
{%
   \closing\code
   \closing\code
}
```

ok

```
\newenvironment{env}[1]
{%
   \opening\code
   \opening\code
}{%
   \closing\code
   \closing\code
}
```

Ouch!

```
\newenvironment{env}[1]{%
   \opening\code
   \opening\code}{%
   \closing\code
   \closing\code}
```

ok

```
\newenvironment{env}[1]{%
   %% \begin{env}{opt}
   \opening\code
   \opening\code}{%
   %% \end{env}
   \closing\code
   \closing\code}
```

- Note: brace position may require eol `%` char

Exceptional situations / oddities

No rule without exception

Forced indentation

```
\@ifnextchar[%] syntax screwup!
  {\@fxbeginsenv{#2}}{\@@fxbeginsenv{#2}}}
```

```
\@ifnextchar[%] syntax screwup!
  {\@fxbeginsenv{#2}}
  {\@@fxbeginsenv{#2}}}
```

Empty body-like macro arguments

```
\@ifundefined{#1note}{}{%
  \@fxpkgerror{command prefix '#1' already in use}{%
    You have called \string\FXRegisterAuthor\space with a command prefix
    already in use.\MessageBreak
    Please choose another one.}}
```

How maniac can you be?
Inter-macro indentation

```
\newcommand\text{%
  \@nextentry
  \noalign\bgroup
    \gdef\@beforespace{\subrubricbeforespace}%
    \@ifstar{\@stext}{\@text}}

\newcommand\@text[1]{%
    \gdef\@nextentry{}%
  \egroup% end of \noalign opened in \text.
  \multicolumn{3}{@{}p{\linewidth}@{}}{\@rubrictextfont #1}\\}

\newcommand\@stext{%
    \gdef\@nextentry{\egroup\\\par}%
  \egroup% end of \noalign opened in \text.
  \multicolumn{3}{@{}p{\linewidth}@{}}\bgroup\@rubrictextfont}
```

Names

You get one for life, so beware. . .

1. Use prefixes
   - Avoid name clashes (*e.g.* `\text` in CurVe and `siunitx`)
   - Mandatory for styles, arguable for classes
   - Use one and stick to it!
     (`\finkdir` *vs.* `\fnk@maindir`)
2. Use postfixes (beware the `\new*` commands!)
   - `\newsavebox\myitemsBOX`
     vs. `\newcounter{myitems}`
3. From the Companion
   - Lowercase for API
   - Mixed case for extension API
   - `@` character for internals (several levels)
4. But stop the `m@dness`!
   - `\@latexerr`, `\@latex@error`
   - `\@input`, `\@@input`, `\@input@`, `\@filef@und`
   - `\sixt@@n`, `\g@addto@macro`

Examples

API

```
\fxnote
\fxuselayout

\FXLayoutInline
\FXRegisterAuthor
```

Internals

```
\@fxnote
\@fxuselayout

\@FXLayoutInline
\@FXRegisterAuthor
```

Nesting levels

```
\DeclareRobustCommand\fxnote{%
  %% ...
  \@ifstar{%
    %% \fxnote*
    \@ifnextchar[%]
      {\@fxsnote{#2}}{\@@fxsnote{#2}}}{%
    %% \fxnote
    \@ifnextchar[%]
      {\@fxnote{#2}}{\@@fxnote{#2}}}}

\long\def\@fxsnote#1[#2]#3#4{%
  %% ...
  \@@fxsnote{#1}{#3}{#4}}

\long\def\@@fxsnote#1#2#3{%
  \implement\me}
```

Polymorphic macros

```
\def\@@@fxnote@early@draft{\for\draft\mode}
\def\@@@fxnote@early@final{\for\final\mode}
%% ...
\let\@@@fxnote@early\@@@fxnote@early@final
```

Exceptional situations / oddities

No exception without rule

- Conforming to *de facto* standards
  - `\ifmycondition`
  - `\listoffixmes`, `\listfixmename`
  - But `\fixmeindexname` or `\fxindexname` ?
- Forced exceptions
  - Manual: `\l@fixme`
  - Auto: `\c@mycounter`, `\myenv`, `\endmyenv`
- Commands *vs.* environments
  - `\fxnote` but `\begin{anfxnote}\end{anfxnote}`

General design rules

1. Don't reinvent the wheel / Use existing tools
   - `calc`, `ifthen`, `record` (!) *etc.*
   - Higher abstraction $\Longrightarrow$ better readability
2. Duplication is evil / Copy-paste is evil
   - Use wrappers
   - Use abstractions
3. Conditionals are evil
   - Centralize the logic
   - Be polymorphic
4. Be modular
   - Use `docstrip`
   - Write small macros

Duplication is evil / Copy-paste is evil

Use wrappers and abstractions

Bad

```
\define@key[fx]{layout}{morelayout}{%
  ...}
\define@cmdkey[fx]{layout}{innerlayout}{%
  ...}
\define@key[fx]{envlayout}{envlayout}{%
  ...}
```

Good

```
\newcommand\@fxdefinekey{
  \define@key[fx]}
\newcommand\@fxdefinecmdkey{
  \define@cmdkey[fx]}

%% ...

\@fxdefinekey{layout}{morelayout}{%
  ...}
\@fxdefinecmdkey{layout}{innerlayout}{%
  ...}
\@fxdefinekey{envlayout}{envlayout}{%
  ...}
```

Bad

```
\define@boolkey[fx]{lang}{langtrack}
  [true]{}
\@fxdefinevoidkey{lang}{nolangtrack}{%
  \@nameuse{fx@lang@langtrack}{false}}

\define@boolkey[fx]{log}{silent}
  [true]{}
\@fxdefinevoidkey{log}{nosilent}{%
  \@nameuse{fx@log@slient}{false}}
```

Good

```
\newcommand*\@fxdefineboolkey[3][]{%
  \define@boolkey[fx]{#2}{#3}
    [true]{#1}
  \@fxdefinevoidkey{#2}{no#3}{%
    \@nameuse{fx@#2@#3}{false}}}

%% ...

\@fxdefineboolkey{lang}{langtrack}
\@fxdefineboolkey{log}{silent}
```

Conditionals are evil

Centralize the logic, be polymorphic

Bad

```
\newif\ifdraft

\def\do@everything{%
  \ifdraft
    \@dothis\this\way
  \else
    \@dothis\this\other\way
  \fi
  %% ...
  \ifdraft
    \@dothat\that\way
  \else
    \@dothat\that\other\way
  \fi}

\DeclareOption{draft}{
  \ifdrafttrue}
\DeclareOption{final}{
  \ifdraftfalse}
\ExecuteOptions{final}
\ProcessOptions
```

Good

```
\def\@dothis@draft{\this\way}
\def\@dothis@final{\this\other\way}

\def\@dothat@draft{\that\way}
\def\@dothat@final{\that\other\way}

\def\do@everything{%
  \@dothis
  %% ...
  \@dothat}

\DeclareOption{draft}{
  \let\@dothis\@dothis@draft
  \let\@dothat\@dothat@draft}
\DeclareOption{final}{
  \let\@dothis\@dothis@final
  \let\@dothat\@dothat@final}
\ExecuteOptions{final}
\ProcessOptions
```

Be modular
Write small macros

Veeeeeeeeeeery bad, splitbib, veeeeeeeeeeery bad!!

```
\def\NMSB@writeentry##1##2##3##4##5,{\ifx\relax ##5\relax
  \else
    \def\NMSB@currcat{##1##2##3##4}%
    \def\NMSB@currcatlevelone{##1##2}%
    \ifx\NMSB@currcatlevelone\NMSB@prevcatlevelone\else
      \expandafter\ifx\csname NMSBtitle@\NMSB@currcatlevelone
       \endcsname\relax
      \else
        \if@filesw
          \expandafter\let\expandafter\NMSB@tempentry
            \csname NMSBtitle@\NMSB@currcatlevelone\endcsname
          \edef\NMSB@tempentry{\@backslashchar subrubric{\NMSB@tempentry}}
          \expandafter\NMSB@writecatbib\expandafter{\NMSB@tempentry}
          \expandafter\ifx\csname
           NMSBcomment@\NMSB@currcatlevelone\endcsname\relax
          \else
            \immediate\write\NMSB@catbib{\string\vskip2ex^^J%
              \string\hspace{-\leftmargin}\string\relax^^J%
              \string\begin{minipage}{\textwidth}^^J%
              \string\addtolength\string\parindent{20pt}^^J%
              \string\noindent}
            \expandafter\let\expandafter\NMSB@tempentry
              \csname NMSBcomment@\NMSB@currcatlevelone\endcsname
            \expandafter\NMSB@writecatbib\expandafter{%
              \NMSB@tempentry^^J}%
            \immediate\write\NMSB@catbib{\string\end{minipage}^^J%
              \string\par\string\addpenalty{\NMSB@penalty}
              \string\vskip2ex}
          \fi
        \fi
      \fi
      \xdef\NMSB@prevcatlevelone{##1##2}%
      \ifnum\theSBresetdepth>0\relax
        \setcounter{\@listctr}{0}%
      \fi
      \expandafter\ifx\csname NMSBprefix@\NMSB@prevcatlevelone
       \endcsname\relax
        \NMSB@currprefixlevelonetok{\relax}%
      \else
        \expandafter\expandafter\expandafter\NMSB@currprefixlevelonetok
        \expandafter\expandafter\expandafter{%
           \csname NMSBprefix@\NMSB@prevcatlevelone\endcsname}%
      \fi
    \fi
```

```
\ifx\NMSB@currcat\NMSB@prevcat\else
  \ifnum\NMSB@currcat=9999\else
    \expandafter\ifx\csname NMSBtitle@\NMSB@currcat\endcsname
     \relax
    \else
      \if@filesw
        \immediate\write\NMSB@catbib{%
          \string\par\string\addpenalty{-\NMSB@halfpenalty}%
          \string\relax^^J%
          \string\item[]%
          \string\SBsubtitle}%
        \expandafter\let\expandafter\NMSB@tempentrya
          \csname NMSBtitle@\NMSB@currcatlevelone\endcsname
        \expandafter\let\expandafter\NMSB@tempentryb
          \csname NMSBtitle@\NMSB@currcat\endcsname
        \expandafter\NMSB@writecatbib\expandafter{%
          \expandafter{\NMSB@tempentrya}}
        \expandafter\NMSB@writecatbib\expandafter{%
          \expandafter{\NMSB@tempentryb}}
        \immediate\write\NMSB@catbib{\string\relax^^J%
          \string\par\string\addpenalty{\NMSB@penalty}%
          \string\relax}%
        \expandafter\ifx\csname
         NMSBcomment@\NMSB@currcat\endcsname\relax
        \else
          \immediate\write\NMSB@catbib{\string\vskip2ex^^J%
            \string\hspace{-\leftmargin}\string\relax^^J%
            \string\begin{minipage}{\textwidth}^^J%
            \string\addtolength\string\parindent{20pt}^^J%
            \string\noindent}
          \expandafter\let\expandafter\NMSB@tempentry
            \csname NMSBcomment@\NMSB@currcat\endcsname
          \expandafter\NMSB@writecatbib\expandafter{%
            \NMSB@tempentry^^J}%
          \immediate\write\NMSB@catbib{\string\end{minipage}^^J%
            \string\par\string\addpenalty{\NMSB@penalty}%
            \string\vskip2ex}
        \fi
      \fi
    \fi
  \fi
  \xdef\NMSB@prevcat{##1##2##3##4}%
  \ifnum\theSBresetdepth>1\relax
    \setcounter{\@listctr}{0}%
  \fi
  \expandafter\ifx\csname NMSBprefix@\NMSB@currcat\endcsname\relax
    \expandafter\NMSB@currprefixtok\expandafter{%
       \the\NMSB@currprefixlevelonetok}%
  \else
    \expandafter\expandafter\expandafter\NMSB@currprefixtok
    \expandafter\expandafter\expandafter{%
      \csname NMSBprefix@\NMSB@currcat\endcsname}
  \fi
\fi
```

```
\expandafter\ifx\csname NMSBlabel@##5\endcsname\relax
  \if@filesw
    \stepcounter\@listctr
    \expandafter\ifx\expandafter\relax\the\NMSB@currprefixtok
      \def\NMSB@tempentry{}%
    \else
      \expandafter\expandafter\expandafter\def
      \expandafter\expandafter\expandafter\NMSB@tempentry
      \expandafter\expandafter\expandafter{%
        \expandafter\the\expandafter\NMSB@currprefixtok
                \the\value{\@listctr}}%
      \edef\NMSB@tempentry{[\NMSB@tempentry]}
    \fi
    \edef\NMSB@tempentry{%
      \@backslashchar bibitem%
      \NMSB@tempentry{\csname NMSBkey@##5\endcsname}}%
    \expandafter\NMSB@writecatbib\expandafter{%
      \NMSB@tempentry}
    \expandafter\let\expandafter\NMSB@tempentry
      \csname NMSBentry@##5\endcsname%
    \expandafter\NMSB@writecatbib\expandafter{%
      \NMSB@tempentry}
  \fi
  \setbox\@tempboxa=\hbox{\the\NMSB@currprefixtok\the\value{\@listctr}}%
  \ifdim\NMSB@reallylongest<\wd\@tempboxa
    \setlength{\NMSB@reallylongest}{\wd\@tempboxa}%
    \xdef\NMSB@reallylongestlabel{%
      \expandafter\ifx\expandafter\relax\the\NMSB@currprefixtok
\else\the\NMSB@currprefixtok\fi\the\value{\@listctr}}%
  \fi
\else
  \if@filesw
    \immediate\write\NMSB@catbib{%
      \string\bibitem}
    \expandafter\let\expandafter\NMSB@tempentry
      \csname NMSBlabel@##5\endcsname
    \expandafter\NMSB@writecatbib\expandafter{%
      \expandafter[\NMSB@tempentry]}%
    \immediate\write\NMSB@catbib{%
      {\csname NMSBkey@##5\endcsname}}%
    \expandafter\let\expandafter\NMSB@tempentry
      \csname NMSBentry@##5\endcsname%
    \expandafter\NMSB@writecatbib\expandafter{%
      \NMSB@tempentry^^J^^J}
  \fi
  \setbox\@tempboxa=\hbox{\csname NMSBlabel@##5\endcsname}%
  \ifdim\NMSB@reallylongest<\wd\@tempboxa
    \setlength{\NMSB@reallylongest}{\wd\@tempboxa}%
    \expandafter\let\expandafter\NMSB@reallylongestlabel\csname
NMSBlabel@##5\endcsname
  \fi
\fi
\fi
}
```

- Originally 203 LoC. 156 after dead branches removal. Only interested in the "green" lines…

Behavior

1. Be nice to your users (incl. yourself)
   - Document your packages *properly*
   - Be backward-compatible
   - Use key/value interfaces
2. Be nice to your hackers (incl. yourself)
   - Be bottom-up
   - Organize your code by feature
3. Intercession management
   - Localize behavior
   - `filehook` is crucial

Key/Value interfaces
How do I choose one? Yeah, I know…

- Package level
- `\mysetup` macro
- Macro level

xkeyval example

```
\ExecuteOptionsX[my]<fam1,fam2,...>{opt1=def1,opt2=def2,...}
\ProcessOptionsX*[my]<fam1,fam2,...>

\newcommand*\mysetup[1]{\setkeys[my]{fam1,fam2,...}{#1}}

\newcommand\mymacro[2][]{%
  \setkeys[my]{fam1,fam2,...}{#1}
  ...}
```

Behavior

1. Be nice to your users (incl. yourself)
   - Document your packages *properly*
   - Be backward-compatible
   - Use key/value interfaces
2. Be nice to your hackers (incl. yourself)
   - Be bottom-up
   - Organize your code by feature
3. Intercession management
   - Localize behavior
   - `filehook` is crucial

Standard interface too limited

The LaTeX developer's worst nightmare

- `\@ifpackageloaded`, `\@ifclassloaded`
  - Curative (*a posteriori*) code only
  - What about precautionary code?
- `\AtBeginDocument`
  - Massively defer code execution
  - What about the order?
- Example:
  - Style S calls `\AtBeginDocument{\things}`
  - Class C loads style S
  - How does C intercede on `\things`?

filehook is crucial

Before and after hooks

- Start with your default behavior
- Rewrite on demand and locally
- Example: how *CurVe* handles bibliography

Collaboration and Reactivity

One year and 38 weeks later…

1. Propose collaboration
   - Don't keep it for yourself
   - Don't reinvent the wheel
2. Accept collaboration
   - Be reactive
     - Review and accept patches
     - Examine and implement ideas
   - Open development
     - Use collaborative tools
     - Trust people

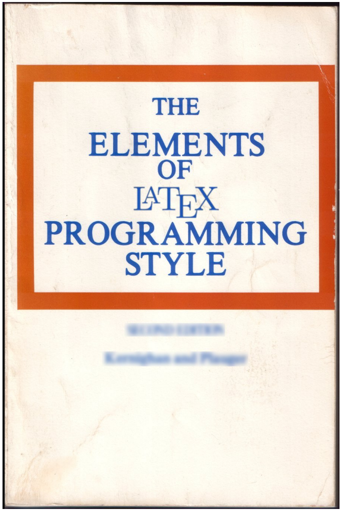
Conclusion

The golden rules of (LaTeX) programming

1. Coding style is important
2. Sticking to it is *more* important
3. Keep it in mind permanently
4. Let it evolve
5. No rule without exception

Perspectives

Where is this book?


THE
ELEMENTS
OF
LaTeX
PROGRAMMING
STYLE

SECOND EDITION

Kernighan and Plauger