

The `latexdemo` package

Matthias Pospiech
matthias@pospiech.eu

v0.1 from 2014/01/30

1 Introduction

In order to demonstrate L^AT_EX code it is very useful to have the code and the resulting output together in the same document. `latexdemo` is a package that provides configurable tools to print out L^AT_EX code and the resulting output in the same document.

The difference to other similar packages is the support of verbatim material inside a conditional sequence and the consequence that each code must be written to an external file.

The commands provided by this package are based on the packages `listings`, `mdframed` and `filecontents`.

2 Basic example

Below is a principle example, which demonstrates the usage of the package by showing some commands of a package (`soul`) with the resulting code side by side:

Code:

```
\so{letterspacing}, \\  
\ul{underlining},   \\  
\st{overstriking}   \\  
and \hl{highlighting}.
```

Result:

```
l e t t e r s p a c i n g,  
u n d e r l i n i n g,  
o v e r s t r i k i n g  
and highlighting.
```

which is created with the code of this package using `\ifcsdef` from `etoolbox` to test if the code will run through or fail because of unknown commands.

```
\begin{filecontents*}{\democodefile}  
\so{letterspacing}, \\  
\ul{underlining},   \\  
\st{overstriking}   \\  
and \hl{highlighting}.  
\end{filecontents*}
```

```

\ifcsdef{so}{%
%
\PrintDemo{style=parallel}
%
}{%
\DemoError{Command \cs{so} of package%
\package{soul} not available.
Probably the package was not loaded.
}
}%

```

3 Usage

3.1 Define code

This package requires the example code to be written to an external file. The filename is saved in the command sequence `\democodefile`. The output is done with the `filecontents` environment:

```

\begin{filecontents*}{\democodefile}
... code ...
\end{filecontents*}

```

The requirement for an external file originates from the problem that verbatim content cannot be saved in normal tex macros since the line breaks and white spaces get lost. Furthermore, any solution which would solve this problem would fail finally because saving such content cannot be included in conditional code statements¹.

The content in the file defined by `\democodefile` is further read for the printing of the code and the corresponding output.

3.2 Print code and result

`\PrintDemo` {style=*<option>*}

This is the macro for the output of code and result. The layout of both is defined with the style option. The following options are possible

parallel	code and result side by side.
stacked	(default) code and result with 100% text width stacked.
lines	like stacked , but with lines on top and bottom of the result instead of a surrounding box.
none	like stacked , but with nothing around the result.
page	result on a single page.

¹See discussion on: <http://tex.stackexchange.com/questions/29256/>

You should use `parallel` for small examples and `stacked` otherwise. The options `lines` and `none` is primarily for those cases where a surrounding box is disturbing or impossible. The latter occurs for example in cases where content is written across the text width boundaries. The option `page` is obviously for those cases where the output is very large or written to another page anyway.

3.2.1 Examples

The following code is used in the examples

```
\begin{filecontents*}{\democodefile}
This code shows some basic math: $a^2 + b^2 = c^2$.
\end{filecontents*}
```

- `\PrintDemo{style=parallel}`

Code:

This code shows some basic math:
 $a^2 + b^2 = c^2$.

Result:

This code shows some basic math:
 $a^2 + b^2 = c^2$.

- `\PrintDemo{style=stacked}`

Code:

This code shows some basic math: $a^2 + b^2 = c^2$.

Result:

This code shows some basic math: $a^2 + b^2 = c^2$.

- `\PrintDemo{style=lines}`

Code:

This code shows some basic math: $a^2 + b^2 = c^2$.

Result:

This code shows some basic math: $a^2 + b^2 = c^2$.

-
- `\PrintDemo{style=none}`

Code:

This code shows some basic math: $a^2 + b^2 = c^2$.

Result:

This code shows some basic math: $a^2 + b^2 = c^2$.

The prefix text for code and result and the filename can be changed. The commands are introduced in section 3.3. Commands for the output of content such as the name of a package, a command, an environment or a generalized error message are introduced in section 3.4.

3.3 Setup

The following commands define the name for the temporary file or the strings used for the printing of code or results. Use `\renewcommand` to change the definitions.

<code>\democodefile</code>	Filename for the temporary file required for code and results printing.
	Default: <code>democode</code>
<code>\democodeprefix</code>	Prefix text for output of code.
	Default: <code>Code:</code>
<code>\demoresultprefix</code>	Prefix text for output of the result.
	Default: <code>Result:</code>

3.4 Output commands

The following commands are provided for the user to print out and format commands, environments, packages and errors. Some are provided by the `doctools` package.

<code>\command</code>	<code>{\langle cmd \rangle}</code> Prints out the argument <code>\command{foo}</code> as <code>\foo</code> .
<code>\cs</code>	<code>{\langle cmd \rangle}</code> Shortcut for <code>\command</code> .
<code>\arg</code>	<code>{\langle cmd \rangle}</code> Prints out an argument in curled brackets without the use of angle brackets as in <code>\marg</code> or <code>\oarg</code> . Thus prints <code>\arg{foo}</code> as <code>{foo}</code> .
<code>\environment</code>	<code>{\langle environment \rangle}</code> Prints out an environment name as <code>environment</code> .
<code>\env</code>	<code>{\langle environment \rangle}</code> Shortcut for <code>\environment</code>
<code>\package</code>	<code>{\langle package \rangle}</code>

`\DemoError` Prints out a package name as `package`.
prints out the given error message
Example: `Error: foo`

4 Implementation

```
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{latexdemo}[2012/12/01 v0.1 typeset code and resulting
  output]
4 \ifpackageloaded{hypdoc}
5   {\RequirePackage[loadHyperref=true,%
6     createIndexEntries=false,%
7     applyLayout=false]{doctools}}
8 \else\ifpackageloaded{doc}
9   {\RequirePackage[loadHyperref=false,%
10    createIndexEntries=false,%
11    applyLayout=false]{doctools}}
12   {}
13 %%% listings (must be loaded before \AtBeginDocument)
14 \RequirePackage{listings}
15 \PassOptionsToPackage{table}{xcolor}
16 % This code needs to be executed at the beginning
17 % of the document because some packages (eg. xcolor)
18 % could lead to option clashes otherwise
19 %
20 %%% Programming
21 \RequirePackage{xspace}
22 \RequirePackage{etoolbox}
23 %%% Write contents to files
24 \RequirePackage{filecontents}
25 %%% Packages for frames
26 \RequirePackage{mdframed}
27 \RequirePackage{framed}
28 %
29 \AtBeginDocument{%
30 %
```

4.1 Preamble

4.1.1 Packages

```
31 %%% Package for colors
32 \RequirePackage{xcolor}
33 %
34 %%% load doctools without hyperref if not loaded and no documentation
35 %%% package was loaded.
36 \ifpackageloaded{doctools}{}
37   {\RequirePackage[loadHyperref=false,%
38     createIndexEntries=true,%
39     applyLayout=false]{doctools}}
40 %
```

4.1.2 Colors

```

41 %% Colors
42 \colorlet{demo@stringcolor}{green!40!black!100}
43 \colorlet{demo@commentcolor}{green!50!black!100}
44 \colorlet{demo@numbercolor}{white!50!black!100}
45 \colorlet{demo@codebackcolor}{white!95!black!100}
46 \colorlet{demo@resultbackcolor}{white}
47 \definecolor{demo@keywordcolor}{rgb}{0,0.47,0.80}
48 \definecolor{demo@rulecolor}{rgb}{0,0.4,0.5}
49 \definecolor{demo@code@rulecolor}{rgb}{0.5,0.5,0.5}
50 %

```

4.2 Commands

```

51 %% === Simple Commands =====
52 %

```

\democodefile Saves the filename for temporary file output.

```

53 \newcommand{\democodefile}{democode}
54 %

```

\democodeprefix Prefix text for code output.

```

55 \newcommand{\democodeprefix}{Code: }
56 %

```

\demoresultprefix Prefix text for result output.

```

57 \newcommand{\demoresultprefix}{\noindent Result:}
58 %

```

\DemoError Output and formatting of error messages.

```

59 %% Print Error
60 \newcommand{\DemoError}[1]{
61   \ifcsdef{textcolor}
62     {\textcolor{red}{Error:~}}
63     {Error:~}
64   #1 \par\noindent
65 }
66 %

```

4.3 Define keys

```

67 %% === Define Keys =====
68 \RequirePackage{kvoptions}
69 \RequirePackage{kvoptions} % options
70 \RequirePackage{pdftexcmds} % string comparison
71 \SetupKeyvalOptions{family=demo,prefix=demo@}
72 %

```

Define default option for style key: *stacked*

```
73 \DeclareStringOption[stacked]{style}
74 \ProcessKeyvalOptions{demo}
75 %
```

`\PrintDemoUsingKeys` Evaluate key and execute corresponding commands

```
76 \newcommand{\PrintDemoUsingKeys}{%
77   \ifnum\pdf@strcmp{\demo@style}{parallel}=0%
78     \PrintCodeAndResultsParallel%
79   \else\ifnum\pdf@strcmp{\demo@style}{stacked}=0%
80     \PrintCodeAndResultsStacked%
81   \else\ifnum\pdf@strcmp{\demo@style}{lines}=0%
82     \PrintCodeAndResultsStackedLines%
83   \else\ifnum\pdf@strcmp{\demo@style}{page}=0%
84     \PrintCodeAndResultsPage%
85   \else\ifnum\pdf@strcmp{\demo@style}{none}=0%
86     \PrintCodeAndResultsNone%
87   \else%
88     \PackageError{latexdemo}{%
89       \MessageBreak%
90       value >\tplbugs@style< unkown \MessageBreak%
91     }{}%
92   \fi\fi\fi\fi\fi%
93 }%
94 %
```

`\PrintDemo` Print code and result using the key-value syntax

```
95 \newcommand{\PrintDemo}[1]{%
96   \begingroup
97     \setkeys{demo}{#1}%
98     \PrintDemoUsingKeys
99   \endgroup
100 }
101 %
```

4.4 listings package style

```
102 %% == Listings style =====
103 %% reuses style from doctools
104 \lstdefinestyle{demostyle}{
105   ,style=lstDemoStyleLaTeXCode%
106   ,numbers=none%
107 }
108 \lstloadlanguages{[LaTeX]TeX}
109 %
```

4.5 mdframed package style


```

110 %% === Mdframed style =====
111 \mdfdefinestyle{DemoStyleFrames}{
112   linecolor=demo@rulecolor,%
113   linewidth=0.8pt,
114   skipabove=0.5\baselineskip,
115   skipbelow=0.5\baselineskip,
116   leftmargin =-3.5pt,
117   rightmargin=-3.5pt,
118   innerleftmargin=3pt,
119   innerrightmargin=3pt,
120   needspace=3\baselineskip,
121 }%
122 %

```

4.6 Commands for the formatting

`\preResultSkip` Default skip at the beginning of a result

```

123 %% === Formating commands =====
124 \newcommand{\preResultSkip}{%\vspace*{-0.5\baselineskip}
125 %

```

`latexresult` Environment to print the result in a box

```

126 \newenvironment{latexresult}{%
127   \demoresultprefix
128   \nopagebreak[4]
129   \preResultSkip
130   \mdfamed[%
131     style=DemoStyleFrames,
132     backgroundcolor=demo@resultbackcolor,%
133     usetwoside=false,
134   ]%
135 }{
136   \endmdfamed
137   \noindent
138 }
139 %

```

`\resultline` Single Line for results

```

140 \newcommand{\resultline}{%
141   \nopagebreak[4]
142   %% Insert single line
143   \mdfamed[%
144     style=DemoStyleFrames,
145     skipabove=3pt,
146     skipbelow=3pt,
147     topline=true,bottomline=false,leftline=false,rightline=false,
148     backgroundcolor=white,%
149   ]\mbox{}\endmdfamed

```

```

150 \nopagebreak[4]
151 }
152 %

```

4.7 Low level commands for printing of code and result

`\printlatexcode` Prints the code using `\lstinputlisting`

```

153 %% === Output commands =====
154 %% Print Code with prefix
155 \newcommand{\printlatexcode}[1][\democodefile]{%
156 \def\demoInputFile{#1}%
157 \IfFileExists{\demoInputFile.tex}{%
158 \democodeprefix%
159 \lstinputlisting[style=demostyle,nolol=true]{\demoInputFile}}{%
160 }%
161 %

```

`\printlatexresult` Prints the result enclosed in the `latexresult` environment. The evaluation of the code is simply achieved by loading the file with `\input`.

```

162 %% Print Result with standard box
163 \newcommand{\printlatexresult}[1][\democodefile]{%
164 \def\demoInputFile{#1}%
165 \begin{latexresult}%
166 \IfFileExists{\demoInputFile.tex}{\input{\demoInputFile.tex}}{%
167 \end{latexresult}%
168 }%
169 %

```

`\printlatexresultlines` Like `\printlatexresult` but with lines above and below instead of a surrounding box.

```

170 %% Print result with lines
171 \newcommand{\printlatexresultlines}{%
172 \demoresultprefix
173 \nopagebreak[4] \resultline \nopagebreak[4]
174 \IfFileExists{\democodefile}{\input{\democodefile}}{%
175 \nopagebreak[4] \resultline \nopagebreak[4]
176 }%
177 %

```

4.8 Output of code and result

```

\PrintCodeAndResultsParallel 178 %% === Output commands for code and result =====
179 \newcommand{\PrintCodeAndResultsParallel}{%
180 \nopagebreak[4]
181 \vspace*{0.5em}\par\noindent

```

```

182 \begin{minipage}[t]{0.48\linewidth}
183 \printlatexcode
184 \end{minipage} \hfill
185 \begin{minipage}[t]{0.48\linewidth}
186 \printlatexresult
187 \end{minipage}
188 \par\noindent
189 }
190 %

```

```

\PrintCodeAndResultsStacked 1 \newcommand{\PrintCodeAndResultsStacked}{%
192 \nopagebreak[4]
193 \vspace*{0.5em}\par\noindent
194 \printlatexcode%
195 \printlatexresult%
196 \par\noindent
197 }%
198 %

```

```

PrintCodeAndResultsStackedLines 99 \newcommand{\PrintCodeAndResultsStackedLines}{%
200 \nopagebreak[4]
201 \vspace*{0.5em}\par\noindent
202 \printlatexcode%
203 \printlatexresultlines%
204 \vspace*{0.5em}\par\noindent
205 }%
206 %

```

```

\PrintCodeAndResultsNone 97 \newcommand{\PrintCodeAndResultsNone}{%
208 \nopagebreak[4]
209 \vspace*{0.5em}\par\noindent
210 \printlatexcode%
211 %

212 \demoresultprefix
213 \nopagebreak[4]
214 \par\noindent
215 \IfFileExists{\democodefile}{\input{\democodefile}}{}%
216 %

217 \vspace*{0.5em}\par\noindent
218 }%
219 %

```

```

\PrintCodeAndResultsPage 220 \newcommand{\PrintCodeAndResultsPage}{%
221 \nopagebreak[4]
222 \vspace*{0.5em}\par\noindent
223 \printlatexcode%
224 \demoresultprefix: Shown on the following page.
225 \newpage
226 \IfFileExists{\democodefile}{\input{\democodefile}}{%
227 \newpage
228 }%
229 %

230 } % end of \AtBeginDocument
231 %

```

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the definition; numbers in *roman* refer to the pages where the entry is used.

A		E	\PrintCodeAndResultsStacked
\arg	<i>4</i>	\env	<i>4</i>
C		\environment	<i>4</i>
\command	<i>4</i>	\PrintCodeAndResultsStackedLines	
\cs	<i>4</i> <i>11</i>	
D		P	\PrintDemo
\democodefile	<i>4</i> , <i><u>7</u></i>	\package	<i>2</i> , <i><u>8</u></i>
\democodeprefix ..	<i>4</i> , <i><u>7</u></i>	\preResultSkip	\PrintDemoUsingKeys . <i><u>8</u></i>
\DemoError	<i>5</i> , <i><u>7</u></i>	\PrintCodeAndResultsNone	\printlatexcode ... <i><u>10</u></i>
\demoresultprefix	<i>4</i> , <i><u>7</u></i> <i><u>11</u></i>	\printlatexresult . <i><u>10</u></i>
		\PrintCodeAndResultsPage	\printlatexresultlines
	 <i><u>11</u></i> <i><u>10</u></i>
		\PrintCodeAndResultsParallel	R
	 <i><u>10</u></i>	\resultline
			<i>9</i>