The schl package*

Tassos Tsesmetzis

ttsesmetzis@gmail.com

November, 2019

Abstract

schl is a XeLaTeX package that provides commands and environments suitable for document types that appear in a classroom enviromnent. It's development is based on the Greek school practice, but it may be usefull in other contexts also.

1 Introduction

Worksheets and tests are common document types in a classroom. schl package comes with macros that facilitate the creation of these documents. It has list environments for questions, exercises and tasks. Other environments of the package can be used for tickable or multiple choice answers. There are also commands for typesetting solutions, hints and answers to exercises.

Furthermore, you can set the name of the teacher, subject, grade, headmaster, school, date, school year and use these to print school's logo or information about an exam. schl has commands to typeset headers for each document type, a macro for typing the points of an exercise and two commands for blank space. There is also a macro for typesetting a wish for good luck!

By default, schl prints all macros that accept text as undefined. As of this version (v1.0), Greek is the only supported language. You can set it with the option greek. Other languages can be supported by redefining package's internal macros.

schl loads the packages fontspec, enumitem, mdframed and amsmath. It is written for XeLaTeX, but can be used by any system that supports fontspec.

2 Macros

2.1 Blank space

\lowerdots
\blankspace

Usually, we need to designate blank space in a document. schl package has two

*This document corresponds to schl v1.0, dated 2019/11/07.

commands for this. The first one `\lowerdots [⟨length⟩]{⟨number⟩}`, prints ⟨*number*⟩ dots. Optional argument ⟨*length*⟩ sets the deviation from base line. It's default value is `-0.3ex`.

```
\newcommand\lowerdots[2][-0.3ex]{%
  \begingroup
  \lccode`m=`.\relax
  \raisebox{#1}{\lowercase\expandafter{\romannumeral\number\number#2 000}}%
  \endgroup
}
```

`\blankspace [⟨length⟩]{⟨linelength⟩}` prints a line with length ⟨*linelength*⟩. The optional argument is the deviation from the base line and it's default value is `-0.3ex`. `\schl@rulethickness` is the default thickness for all `\blankspace` lines.

```
\newcommand\blankspace[2][-0.3ex]{%
 \raisebox{#1}{\rule{#2}{\schl@rulethickness}}
}
```

2.2 Lists

`schl` package defines seven types of lists. These are `question`, `exercise`, `schltask`, `multichoice`, `tickchoice`, `truefalse` and `matchique`. `tickchoice` comes also with a stared version `tickchoice*`. All of them depend on the package `enumitem`.

`question` `exercise` `schltask`

These environments are `enumerate`-like lists. List's `\item` is of the form ⟨*type*⟩ ⟨*counter*⟩, where `type` is `\question@term` for `question`, `\exercise@term` for `exercise` and `\task@term` for `schltask`. ⟨*counter*⟩ is the internal counter of the environment.

```
\newlist{question}{enumerate}{1}
\setlist*[question]{%
  align=left,
  label=\bf \question@term\  \arabic*.,
  wide,
  leftmargin=0pt,
  labelindent=0pt
}

\newlist{exercise}{enumerate}{1}
\setlist*[exercise]{%
  align=left,
  label=\bf\exercise@term\  \arabic*.,
  wide,
  leftmargin=0pt,
  labelindent=0pt
}

\newlist{schltask}{enumerate}{1}
\setlist*[schltask]{%
  align=left,
  label=\bf\letterspace{\defaultletterspace}\task@term\  \arabic*,
  wide,
  leftmargin=0pt,
  labelindent=0pt
```

```
}
```

\letterspace

The macro \letterspace{⟨*number*⟩} is used to set the horizontal space of adjacent characters in a word. It is based on the \addfontfeature macro from the package fontspec. The argument ⟨*number*⟩ is a percentage of the font size. In schl package is used to set the space between capital word letters.

```
\def\letterspace#1{\addfontfeature{LetterSpace=#1}}
```

multichoice

The multichoice environment is used to typeset multiple choice answers.

```
\newlist{multichoice}{enumerate*}{1}
\setlist*[multichoice]{
  labelindent=\parindent,
  label=\Alph*.,
  itemjoin=\hspace{\fill},
  before=\hspace{\fill},
  after=\hspace{\fill}
}
```

tickchoice
tickchoice*

The environments tickchoice and tickchoice* are variants of the itemize list. For both cases, each item is preceded by a square. tickchoice stacks items vertically,

```
\newlist{tickchoice}{itemize}{1}
\setlist[tickchoice]{labelindent=\parindent,label={\large$\square$}}
```

while tickchoice* stacks them horizontally.

```
\newlist{tickchoice*}{itemize*}{1}
\setlist*[tickchoice*]{
  labelindent=\parindent,
  label={\large$\square$},
  itemjoin=\hspace{\fill},
  before=\hspace{\fill},
  after=\hspace{\fill}
}
```

truefalse

truefalse is a variant of the enumerate environment. Each \item is divided in two parts. The first part is the text that follows the \item macro. The second part is a \parbox that prints \trueabbr@term and \falseabbr@term.

```
\newlist{truefalse}{enumerate}{1}
\setlist[truefalse]{label={\bf \arabic*.},%
  before*={%
    \let\defaultitem\item%      Save the standard definition of \item in a macro.
    \toggletrue{first}%                 Set the first toggle with initial value true.
    \def\item{%
      \iftoggle{first}{%
        \togglefalse{first}%              Set the first toggle to take the value false.
        \defaultitem\begin{minipage}[t]{0.8\linewidth minus \truefalselength}%
        }{%
        \end{minipage}\hfill\truefalselabel\defaultitem%
        \begin{minipage}[t]{0.8\linewidth minus \truefalselength}%
        }
      }% new, temporary defition of \item
    },
```

```
    after*={%   This takes care of adding the fill for the final item on
      %         the list and just makes sure that \item is reset to its standard definition
    \end{minipage}\hfill\truefalselabel% fill for final item in list
    \let\item\defaultitem% restore standard definition of \item
  }%
}
```

truefalse* `truefalse*` is based on the `truefalse` environment. A line with `\trueabbr@term` and `\falseabbr@term` at the end prepends the list of items. In contrast with the `truefalse` environment, each `\item` ends with a pair of squares.

```
\newlist{truefalse*}{enumerate}{1}
\setlist[truefalse*]{label={\bf \arabic*.},%
  before*={%
    \hfill\truefalselabel%      Print first line only with the terms True - False
    \let\defaultitem\item%      Save the standard definition of \item in a macro.
    \toggletrue{first}%                 Set the first toggle with initial value true.
    \def\item{%
      \iftoggle{first}{%
        \togglefalse{first}%              Set the first toggle to take the value false.
        \defaultitem\begin{minipage}[t]{0.8\linewidth minus \truefalselength}%
        }{%
        \end{minipage}\hfill\truefalsesquares\defaultitem%
        \begin{minipage}[t]{0.8\linewidth minus \truefalselength}%
        }
      }% new, temporary defition of \item
    },
    after*={%   This takes care of adding the fill for the final item on
      %         the list and just makes sure that \item is reset to its standard definition
    \end{minipage}\hfill% fill for final item in list
    \truefalsesquares
    \let\item\defaultitem% restore standard definition of \item
  }%
}
```

matchingque The macro `\matchingque{`$\langle CSV \rangle$`}{`$\langle CSV \rangle$`}` is used to typeset matching questions. $\langle CSV \rangle$ are comma separated values. The $\langle CSV \rangle$s of the first argument are the parts of the matching questions that will be print in the left column. Similarly, the $\langle CSV \rangle$ of the second argument are going to be printed on the right column of the matching questions.

```
\newcommand\matchingque[3][300pt]{%
  \begin{center}
    \parbox[c]{#1}{
      \parbox[c]{\leftmatchwidth}{%
        \begin{leftmatching}
          \@for\tmp:=#2%
          \do{%
          \item \tmp
          }
        \end{leftmatching}
      }\hfill%
      \parbox[c]{\rightmatchwidth}{%
```

```
        \begin{rightmatching}
          \@for\tmp:=#3%
          \do{%
          \item \tmp
          }
        \end{rightmatching}
      }
    }
  \end{center}
}
```

leftmatching
rightmatching

Environments `leftmatching` and `rightmatching` are used to typeset each column in `\matchingque`.

```
\newlist{leftmatching}{enumerate}{1}
\newlist{rightmatching}{enumerate}{1}
\setlist*[leftmatching]{label=\bf\Alph*.}
\setlist*[rightmatching]{label=\bf\arabic*.}
```

2.3 Answers, solutions and hints

\answer

Macro `\answer{`⟨*text*⟩`}` prints `(\answerabbr@term \meta{text})` at the right end of the current line.

```
\newcommand\answer[2][\hfill\footnotesize]{%
  {#1 (\answerabbr@term: #2)}
}
```

\solution

Macro `\solution{`⟨*text*⟩`}` is used to typeset the solution of an exercise.

```
\newcommand\solution[1]{%
  \par\noindent\phantom{.}\hfill\textbf{\solution@term}\hfill\phantom{.}\par%
  \noindent #1
}
```

\hint

schl provides the macro `\hint{`⟨*text*⟩`}` for typesetting exercise hints.

```
\newcommand\hint[2][\par\noindent\footnotesize]{%
  {#1\textbf{\hint@term:} #2}%
}
```

\deadline

A feature of homework assignments is a deadline date. `\deadline{`⟨*date*⟩`}` prints `\deadline@term` followed by argument ⟨*date*⟩.

```
\newcommand\deadline[2][\noindent\bf]{%
  {#1\deadline@term}: #2%
}
```

\remark

Add a remark in a document. `\remark{`⟨*text*⟩`}` prints `\remark@term` followed by argument ⟨*text*⟩.

```
\newcommand\remark[2][\noindent\bf]{%
  {#1\remark@term}: #2%
}
```

\reminder

Add a reminder in a document. `\reminder{`⟨*text*⟩`}` prints `\reminder@term` followed by argument ⟨*text*⟩.

```
\newcommand\reminder[2][\noindent\bf]{%
  {#1\reminder@term}: #2%
}
```

2.4 Titles and headers

\heading Common document types in a school environment are the worksheet, various tests and final written exams. The macro \heading{⟨*text*⟩} gives a generic header for all these documents.

```
\newcommand\heading[1]{%
  \begin{center}
    {\bf\large #1}
  \end{center}
}
```

\worksheethd Macro \worksheethd{⟨*text*⟩} sets the title of a worksheet. It appends ⟨*text*⟩ to \worksheet@term.

```
\newcommand\worksheethd[1]{%
  \heading{\worksheet@term\  #1}
}
```

\examhd \examhd[⟨*text*⟩]{⟨*text*⟩} is used to set the title of tests. The optional argument has the default value \termtest@term.

```
\newcommand\examhd[2][\termtest@term]{%
  \heading{#1 #2}
}
```

\finalexamhd Titles for end year exams have a standardized form in Greek schools. \exams@term is followed by information about the exam. Then comes \period@term with the exam period after it. \finalexamhd{⟨*info*⟩}{⟨*period*⟩} is used for these cases.

```
\newcommand\finalexamhd[2]{%
  \heading{\letterspace{\defaultletterspace} #1 \exams@term\\[0.5ex] \period@term\  #2}
}
```

\schl@framedbox \schl@framedbox{⟨*text*⟩} prints ⟨*text*⟩ in a centered frame box. It is used by \theorypart and \exercisepart.

```
\newcommand\schl@framedbox[1]{%
  \begin{center}
    \fbox{\large{\bf\letterspace{\defaultletterspace} #1} }%
  \end{center}
}
```

\theorypart \exercisepart Sometimes theory and exercise sections constitute a written test. Macros \theorypart and \exercisepart print headers for those parts.

```
\newcommand\theorypart{%
  \schl@framedbox{\theoryheader@term\!}
}
```

and

```
\newcommand\exercisepart{%
  \schl@framedbox{\exerciseheader@term\!}
}
```

2.5 School information

\school
\headmaster
\teacher
\subject
\grade
\schoolyear
\setdate
\examtime

The macros \school{⟨*text*⟩}, \headmaster{⟨*name*⟩}, \teacher{⟨*name*⟩}, \subject{⟨*text*⟩}, \grade{⟨*text*⟩}, \schoolyear{⟨*year*⟩}, \setdate{⟨*date*⟩}, \examtime{⟨*time*⟩} define and set the value of internal macros.

```
\newcommand\school[1]{\def\schl@school{#1}}
\newcommand\headmaster[1]{\def\schl@headmaster{#1}}
\newcommand\teacher[1]{\def\schl@teacher{#1}}
\newcommand\subject[1]{\def\schl@subject{#1}}
\newcommand\grade[1]{\def\schl@grade{#1}}
\newcommand\schoolyear[1]{\def\schl@schoolyear{#1}}
\newcommand\setdate[1]{\def\schl@date{#1}}
\newcommand\examtime[1]{\def\schl@examtime{#1}}
```

\authorities

In a similar vein, \authorities{⟨*text*⟩} is used to define the internal macro \schl@authorities.

```
\newcommand\authorities[1]{\def\schl@authorities{#1}}
```

2.6 Other macros for tests

\points

\points[⟨*macro*⟩]{⟨*number*⟩} is used to designate the points of an exercise. {⟨*number*⟩} is the number of points for the current exercise, while [⟨*macro*⟩] can be used to control the space just before the points.

```
\newcommand{\points}[2][\hfill\bf\footnotesize]{%
{#1(\points@term{#2}\  #2)}
}
```

\fullname

\fullname{⟨*text*⟩} prints \fullname@term followed by ⟨*text*⟩.

```
\newcommand\fullname[2][\noindent\rmfamily]{%
  {#1\fullname@term}: #2%
}
```

\datefield

Similarly, \datefield{⟨*text*⟩} prints \date@term with ⟨*text*⟩ after it.

```
\newcommand\datefield[2][\noindent\bf]{%
  {#1\date@term}: #2%
}
```

\getdate

\getdate prints \schl@date. The last macro can be set with \setdate.

```
\newcommand\getdate{\schl@date}
```

\duration

\duration{⟨*duration*⟩} prints \duration@term with ⟨*duration*⟩ after it.

```
\newcommand\duration[2][\noindent\bf]{%
  {#1\duration@term}: #2%
}
```

\schoollogo

\schoollogo{⟨*width*⟩} prints \schl@school, \schl@grade, \schl@subject and \schl@teacher. ⟨*width*⟩ is the length of the \parbox.

```
\def\schoollogo#1{%
  \parbox[t]{#1}{%
    \schl@school\\%
    \schl@grade\\%
```

```
    \schl@subject\\%
    \schl@teacher
  }
}
```

`\authoritylogo` `\authoritylogo[`⟨*number*⟩`]` prints `\sch@authorities` and `\schl@school`. Argument ⟨*number*⟩ is a multiplier for `\baselineskip`. This spaces is added above the macro.

```
\newcommand\authoritylogo[1][1.5]{%
  \noindent\parbox[t][\height]{0.4\textwidth}{%
    \centering%

    \vspace{#1\baselineskip}

    {\schl@authorities}

    \vspace{3\lineskip}

    {\small\letterspace{\defaultletterspace}\MakeUppercase{\schl@school}}
  }
}
```

`\signatures` Some types of written tests end with the names of the headmaster and the teacher(s) followed by handwritten signatures. `\signatures[`⟨*role*⟩`]{`⟨*signer(s)*⟩`}` prints ⟨*role*⟩. After it follow the name(s) of the ⟨*signer(s)*⟩, one per line. ⟨*Signer(s)*⟩ is a comma separated list of values. The default value for ⟨*role*⟩ is `\headmaster@term`. `\signatureslength` is the length of the `\signatures` block and `\signatureslineskip` is the length between adjucent lines.

```
\newcommand\signatures[2][\headmaster@term]{%
  \parbox[t]{\signatureslength}{%
    \setlength \baselineskip{\signaturelineskip}
    \begin{center}
      #1%
      \@for\arg:=#2
      \do {%
        \\%
        \arg%
      }
    \end{center}
  }
}
```

`\wish` `\wish` prints `\schl@wish`, a default wish for tests.

```
\newcommand\wish[1][\schl@wish]{%
  \begin{center}
      {\LARGE\bf #1}
  \end{center}
}
```