lpsudoku.sty

v1.0

A style file for typesetting Sudoku logic puzzles

	2	6						
						1	7	
		3	1		6			
	6			5		8		3
		9	2	6	1	7		
5		4		8			6	
			8		4	3		
	4	8						
						9	4	

1	2	6	5	7	8	4	3	9
4	8	5	9	3	2	1	7	6
7	9	3	1	4	6	5	8	2
2	6	1	4	5	7	8	9	3
8	3	9	2	6	1	7	5	4
5	7	4	3	8	9	2	6	1
6	5	2	8	9	4	3	1	7
9	4	8	7	1	3	6	2	5
3	1	7	6	2	5	9	4	8

March 22, 2013

Package author:
Josef Kleber

josef.kleber@gmx.de
https://bitbucket.org/kleberj/logicpuzzle/

1 The puzzle

Well, it's Sudoku – nothing to explain! Fill each row and column with numbers from 1 to 9. Here's a little self-explanatory example:

	2	6						
						1	7	
		3	1		6			
	6			5		8		3
		9	2	6	1	7		
5		4		8			6	
			8		4	3		
	4	8						
						9	4	

1	2	6	5	7	8	4	3	9
4	8	5	9	3	2	1	7	6
7	9	3	1	4	6	5	8	2
2	6	1	4	5	7	8	9	3
8	3	9	2	6	1	7	5	4
5	7	4	3	8	9	2	6	1
6	5	2	8	9	4	3	1	7
9	4	8	7	1	3	6	2	5
3	1	7	6	2	5	9	4	8

```
\begin{center}
  \begin{lpsudoku}
    \setrow{9}{{},2,6,{},{},{},{},{},{}}
    \setrow{8}{{},{},{},{},{},{},1,7,{}}
    \setrow{7}{{},{},3,1,{},6,{},{},{}}
    \setrow{6}{{},6,{},{},5,{},8,{},3}
    \setrow{5}{{},{},9,2,6,1,7,{},{}}
    \setrow{4}{5,{},4,{},8,{},{},6,{}}
    \setrow{3}{{},{},{},8,{},4,3,{},{}}
    \setrow{2}{{},4,8,{},{},{},{},{},{}}
    \setrow{1}{{},{},{},{},{},{},9,4,{}}
    \end{lpsudoku}
    \hspace{1,5cm}
    \begin{lpsudoku}
    \setrow{9}{1,2,6,5,7,8,4,3,9}
    \setrow{8}{4,8,5,9,3,2,1,7,6}
    \setrow{7}{7,9,3,1,4,6,5,8,2}
    \setrow{6}{2,6,1,4,5,7,8,9,3}
    \setrow{5}{8,3,9,2,6,1,7,5,4}
    \setrow{4}{5,7,4,3,8,9,2,6,1}
    \setrow{3}{6,5,2,8,9,4,3,1,7}
    \setrow{2}{9,4,8,7,1,3,6,2,5}
    \setrow{1}{3,1,7,6,2,5,9,4,8}
  \end{lpsudoku}
```

2 Options

width [9.1cm] sets the width of the minipage, in which the grid is typeset. 9 cells of width 1cm plus a litle extra for lines.

scale [1] scales the size of the grid in the minipage. To get a width of 5cm you need to scale by $^5/_9$

fontsize [Large] specifies the size of the numbers next to the grid. Here, the usual LaTeX sizes are used. Possible values: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

title [] sets the title of a puzzle.

titleindent [0cm] defines the indent of the title.

titlewidth [9.1cm] specifies the width of the box the title is set in.

bgcolor [] sets the background color of the grid.

counterstyle [none] defines the counter style. Predefined styles: none, left, right

cvoffset [-24pt] sets the vertical offset of the counters in the margin.

3 Environments

3.1 lpsudoku

\begin{lpsudoku}[⟨*options*⟩]
...
\end{lpsudoku}

The `lpsudoku`[1] environment is the central core of the style file. With the optional argument of the environment, you can reset the options with local scope. Here, a blank grid is created.

4 Commands

4.1 In the grid and around

4.1.1 lpsudokucell

\lpsudokucell{⟨*column*⟩}{⟨*row*⟩}
{⟨*number*⟩}

The command `\lpsudokucell` sets the ⟨*number*⟩ of the grid cell ⟨*column*⟩ ⟨*row*⟩.

4.1.2 setrow

\setrow{⟨*row*⟩}{⟨*csv list*⟩}

The command `\setrow` sets the contents of ⟨*row*⟩. It expects a comma-separated list.

4.1.3 setcolumn

\setcolumn{⟨*column*⟩}{⟨*csv list*⟩}

The command `\setcolumn` sets the contents of ⟨*column*⟩.

[1] named `lpsudoku` – as well as the style file – to avoid naming conflict with already existing `sudoku.sty`

4.2 Presentation

4.2.1 definecounterstyle

\definecounterstyle{⟨*name*⟩}{⟨*definition*⟩}

The command \definecounterstyle allows you to define your own styles. For example, the style left is defined as follows:

```
\definecounterstyle{left}{
  \begingroup\reversemarginpar\marginnote{
  \tikz\node[shape=rectangle,fill=yellow!40,inner sep=7pt,
             draw,rounded corners=3pt,thick]
  {\Huge\puzzlecounter};}[\LP@cvoffset]\endgroup}
}
```

To typeset the counter into the margin we use the command \marginnote. We need to use the command \reversemarginpar to set the counter into the left margin. Of course, we must use this command in a group for local scope. Finally we use \puzzlecounter in a \tikz node with a vertical offset set with the option cvoffset.

4.2.2 puzzlecounter

\puzzlecounter

The command \puzzlecounter provides the counter in textual form to use it for example in \definecounterstyle.

4.2.3 titleformat

\titleformat{⟨*format*⟩}

With the command \titleformat, you can define the format of the title. By default, the definition is as follows:

```
\titleformat{\centering\Large\color{blue}}
```

4.3 Miscellaneous

4.3.1 lpsudokusetup

\lpsudokusetup{⟨*options*⟩}

With the command \lpsudokusetup you can reset the options with global scope.

4.3.2 setpuzzlecounter

\setpuzzlecounter{⟨*number*⟩}

With the command \setpuzzlecounter, you can reset the puzzle counter, for example before the solutions.

5 Supporting bash scripts

5.1 createlpsudoku

The `createlpsudoku` [Kle13a] `bash` script can transform Sudoku format files into `lpsudoku` environments. It can process files in the so called one line 81 format[2] (option -e (default)) and in simple sudoku format (option -s)

Usage: `createlpsudoku [options] [-o output] -i input`

It expects an input file with the option `-i`. You can specify an output file with the option `-o`. Otherwise it writes to `stdout`. Furthermore, the following options are possible:

-w	write Windows line endings (CR/LF) to file
-v	prints version number
-h	prints help

5.2 lpsmag

With the `lpsmag` [Kle13d] `bash` script you can half automatically produce a Sudoku magazine using the LaTeX package `lpsudoku.sty` and the `createlpsudoku` bash script.

Usage: `lpsmag configfile`

The script needs an installed QQwing [Ost11] and a config file for defining the magazine's contents:

```
page p1 easy
page p2 easy
startpuzzles
typesetpage p1
typesetpage p2
startsolutions
typesetsolpage p1 p2 last
```

This config file will be sourced into the `lpsmag` `bash` script and contains calls of `lpsmag` functions. Make sure, that the config file has UNIX line endings (LF). For a detailed documentation I refer to the following wiki [Kle13d] entry.
After running `lpsmag` you will find a `lpsmag.tex` in your working directory. Just run `pdflatex lpsmag.tex` twice and you finally get for example this `lpsmag.pdf`.

6 Examples & Solutions

You can download application examples and their solutions from the project page [Kle13c].

[2] processing of several sudokus in 81 format (one in each line) is possible

References

[Hob12] Hobiger, Bernhard: *HoDoKu*. 2012. – (http://hodoku.sourceforge.net/en/index.php)

[Kle13a] Kleber, Josef: *createlpsudoku*. 2013. – (http://jklatex.square7.de/wiki/doku.php?id=createlpsudoku)

[Kle13b] Kleber, Josef: *How to create a Sudoku magazine*. 2013. – (http://jklatex.square7.de/wiki/doku.php?id=sudoku_magazine)

[Kle13c] Kleber, Josef: *The logicpuzzle bundle*. 2013. – (https://bitbucket.org/kleberj/logicpuzzle/)

[Kle13d] Kleber, Josef: *lpsmag*. 2013. – (http://jklatex.square7.de/wiki/doku.php?id=lpsmag)

[Ost11] Ostermiller, Stephen: *QQwing – Sudoku Generator and Solver*. 2011. – (http://ostermiller.org/qqwing/)