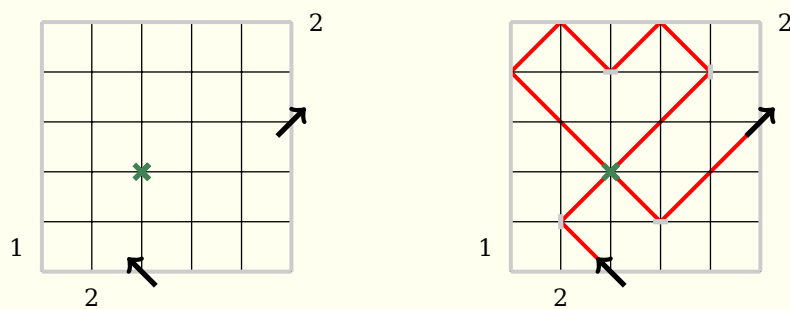


# logicpuzzle.sty

v2.1

A style file for typesetting logic puzzles



May 10, 2013

Package author:  
**Josef Kleber**

1	Supported puzzles	7
1.1	2D-Sudoku	7
1.1.1	Example	7
1.1.2	Options	8
1.2	Battleship	8
1.2.1	Example	9
1.2.2	Options	9
1.3	Bokkusu	10
1.3.1	Example	10
1.3.2	Options	11
1.4	Chaos Sudoku	12
1.4.1	Example	12
1.4.2	Options	13
1.5	Hakuyu	13
1.5.1	Example	14
1.5.2	Options	15
1.6	Hitori	15
1.6.1	Example	16
1.6.2	Options	16
1.7	Kakuro	17
1.7.1	Example	17
1.7.2	Options	18
1.8	Kendoku	18
1.8.1	Example	19
1.8.2	Options	20
1.9	Killer Sudoku	21
1.9.1	Example	21
1.9.2	Options	22
1.10	Laser Beam	23
1.10.1	Example	23
1.10.2	Options	24
1.11	Minesweeper	24
1.11.1	Example	25
1.11.2	Options	25
1.12	Schatzsuche	26
1.12.1	Example	26
1.12.2	Options	27
1.13	Skyline	27
1.13.1	Example	28
1.13.1.1	Variants	28
1.13.1.1.1	Skyline Sudoku	28
1.13.1.1.2	Skyline Sudoku (N*N)	30
1.13.2	Options	31
1.14	Slitherlink	32
1.14.1	Example	32
1.14.2	Options	33

1.15	Sudoku . . . . .	33
1.15.1	Example . . . . .	33
1.15.2	Options . . . . .	34
1.15.3	Supporting bash scripts . . . . .	35
1.15.3.1	createlpsudoku . . . . .	35
1.15.3.2	lpsmag . . . . .	35
1.16	Tents and Trees . . . . .	36
1.16.1	Example . . . . .	36
1.16.2	Options . . . . .	37
1.17	Tunnel . . . . .	37
1.17.1	Example . . . . .	37
1.17.2	Options . . . . .	38
2	Roll out your own grid-based logic puzzle . . . . .	39
3	The code . . . . .	40
3.1	PGF layers . . . . .	40
3.2	Environments . . . . .	41
3.2.1	Puzzle environments . . . . .	41
3.2.1.1	logicpuzzle . . . . .	41
3.2.1.1.1	Options . . . . .	41
3.2.2	Supporting environments . . . . .	42
3.2.2.1	puzzlebackground . . . . .	42
3.2.2.2	puzzleforeground . . . . .	42
3.3	Commands . . . . .	42
3.3.1	Puzzle specific commands . . . . .	42
3.3.1.1	2D-Sudoku . . . . .	42
3.3.1.1.1	ddsudokucell . . . . .	42
3.3.1.1.2	ddsudokusetup . . . . .	42
3.3.1.2	Battleship . . . . .	42
3.3.1.2.1	placeship . . . . .	42
3.3.1.2.2	placesegment . . . . .	43
3.3.1.2.3	ship . . . . .	43
3.3.1.2.4	placewater . . . . .	43
3.3.1.2.5	placeisland . . . . .	43
3.3.1.2.6	shipH . . . . .	43
3.3.1.2.7	shipV . . . . .	43
3.3.1.2.8	shipbox . . . . .	43
3.3.1.2.9	battleshipsetup . . . . .	43
3.3.1.2.10	classicgame . . . . .	43
3.3.1.3	Bokkusu . . . . .	43
3.3.1.3.1	valueH . . . . .	43
3.3.1.3.2	valueV . . . . .	44
3.3.1.3.3	sumH . . . . .	44
3.3.1.3.4	sumV . . . . .	44
3.3.1.3.5	bokkususetup . . . . .	44
3.3.1.4	Chaos Sudoku . . . . .	44
3.3.1.4.1	chaossudokucell . . . . .	44

3.3.1.4.2	chaossudokusetup . . . . .	44
3.3.1.5	Hakuyu . . . . .	44
3.3.1.5.1	hakyuucell . . . . .	44
3.3.1.5.2	hakyuusetup . . . . .	44
3.3.1.6	Hitori . . . . .	44
3.3.1.6.1	hitorisetup . . . . .	44
3.3.1.7	Kakuro . . . . .	44
3.3.1.7.1	kakurorow . . . . .	44
3.3.1.7.2	kakurocolumn . . . . .	45
3.3.1.7.3	KKR . . . . .	45
3.3.1.7.4	Black . . . . .	45
3.3.1.7.5	kakurosetup . . . . .	45
3.3.1.8	Kendoku . . . . .	45
3.3.1.8.1	kendokucell . . . . .	45
3.3.1.8.2	kendokusetup . . . . .	45
3.3.1.9	Killer Sudoku . . . . .	45
3.3.1.9.1	killersudokucell . . . . .	45
3.3.1.9.2	killersudokusetup . . . . .	45
3.3.1.10	Laser Beam . . . . .	45
3.3.1.10.1	laserH . . . . .	45
3.3.1.10.2	laserV . . . . .	45
3.3.1.10.3	mirrorH . . . . .	45
3.3.1.10.4	mirrorV . . . . .	46
3.3.1.10.5	placearrow . . . . .	46
3.3.1.10.6	placecross . . . . .	46
3.3.1.10.7	placemirror . . . . .	46
3.3.1.10.8	laser . . . . .	46
3.3.1.10.9	laserbeamsetup . . . . .	46
3.3.1.11	Minesweeper . . . . .	46
3.3.1.11.1	Mine . . . . .	46
3.3.1.11.2	minesweepersetup . . . . .	46
3.3.1.12	Schatzsuche . . . . .	46
3.3.1.12.1	Diamond . . . . .	46
3.3.1.12.2	schatzsuchesetup . . . . .	46
3.3.1.13	Skyline . . . . .	47
3.3.1.13.1	skylineT . . . . .	47
3.3.1.13.2	skylineB . . . . .	47
3.3.1.13.3	skylineL . . . . .	47
3.3.1.13.4	skylineR . . . . .	47
3.3.1.13.5	skylinecell . . . . .	47
3.3.1.13.6	skylinesetup . . . . .	47
3.3.1.14	Slitherlink . . . . .	47
3.3.1.14.1	slitherlinkcell . . . . .	47
3.3.1.14.2	slitherlinksetup . . . . .	47
3.3.1.15	Sudoku . . . . .	47
3.3.1.15.1	lpsudokucell . . . . .	47
3.3.1.15.2	lpsudokusetup . . . . .	47

3.3.1.16	Tents and Trees	47
3.3.1.16.1	tentH	47
3.3.1.16.2	tentV	48
3.3.1.16.3	Tree	48
3.3.1.16.4	Tent	48
3.3.1.16.5	tentsandtreessetup	48
3.3.1.17	Tunnel	48
3.3.1.17.1	tunnelH	48
3.3.1.17.2	tunnelV	48
3.3.1.17.3	portal	48
3.3.1.17.4	tube	48
3.3.1.17.5	tunnelsetup	48
3.3.2	User commands	48
3.3.2.1	In the grid	48
3.3.2.1.1	setcell	48
3.3.2.1.2	setbigcell	49
3.3.2.1.3	setrow	49
3.3.2.1.4	setcolorrow	49
3.3.2.1.5	setcolumn	49
3.3.2.1.6	setcolorcolumn	49
3.3.2.1.7	setrule	49
3.3.2.1.8	fillcell	49
3.3.2.1.9	fillrow	49
3.3.2.1.10	fillcolumn	49
3.3.2.1.11	filldiagonals	50
3.3.2.1.12	framearea	50
3.3.2.1.13	fillarea	50
3.3.2.1.14	colorarea	50
3.3.2.1.15	framepuzzle	50
3.3.2.1.16	tikzpath	50
3.3.2.2	Presentation	50
3.3.2.2.1	titleformat	51
3.3.2.2.2	puzzlecounter	51
3.3.2.2.3	setpuzzlecounter	51
3.3.2.2.4	definecounterstyle	51
3.3.2.2.5	setgridlinestyle	51
3.3.2.2.6	setnormallinewidth	51
3.3.2.2.7	setthicklinewidth	51
3.3.3	Internal commands	51
3.3.3.1	Initialization	51
3.3.3.1.1	LP@define@key	52
3.3.3.1.2	LP@define@choicakey@fontsize	52
3.3.3.2	Drawing grids	52
3.3.3.2.1	LP@drawgrid	52
3.3.3.2.2	LP@drawsudokugrid	52
3.3.3.2.3	LP@drawbackground	52

3.3.3.3	In the grid . . . . .	52
3.3.3.3.1	LP@setcellcontent . . . . .	52
3.3.3.3.2	LP@setcellcontentC . . . . .	52
3.3.3.3.3	LP@setrowcontents . . . . .	52
3.3.3.3.4	LP@setcolumncontents . . . . .	52
3.3.3.3.5	LP@ingrid . . . . .	53
3.3.3.3.6	LP@definecolor . . . . .	53
3.3.3.4	Around the grid . . . . .	53
3.3.3.4.1	LP@leftcolumn . . . . .	53
3.3.3.4.2	LP@rightcolumn . . . . .	53
3.3.3.4.3	LP@toprow . . . . .	53
3.3.3.4.4	LP@bottomrow . . . . .	53
3.3.3.5	Presentation . . . . .	53
3.3.3.5.1	LP@drawcounter . . . . .	53
4	Examples . . . . .	53
	References . . . . .	55
	Index . . . . .	56

## 1 Supported puzzles

### 1.1 2D-Sudoku

Fill every row, every column and each of the two diagonals – if indicated – with numbers from 1 to SIZE of the grid.

#### 1.1.1 Example

1				
3				4
	4		2	
			3	

1	3	4	5	2
3	2	5	1	4
5	4	3	2	1
2	5	1	4	3
4	1	2	3	5

```

1 \begin{center}
2   \begin{ddsudoku}
3     \framepuzzle
4     \filldiagonals[orange!50]
5     \ddsudokucell{1}{5}{1}
6     \ddsudokucell{1}{4}{3}
7     \ddsudokucell{2}{3}{4}
8     \ddsudokucell{4}{1}{3}
9     \ddsudokucell{4}{3}{2}
10    \ddsudokucell{5}{4}{4}
11  \end{ddsudoku}
12  \hspace{1.5cm}
13  \begin{ddsudoku}
14    \framepuzzle
15    \filldiagonals[orange!50]
16    \setrow{5}{1,3,4,5,2}
17    \setrow{4}{3,2,5,1,4}
18    \setrow{3}{5,4,3,2,1}
19    \setrow{2}{2,5,1,4,3}
20    \setrow{1}{4,1,2,3,5}
21  \end{ddsudoku}
22 \end{center}

```

### 1.1.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.1cm] 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

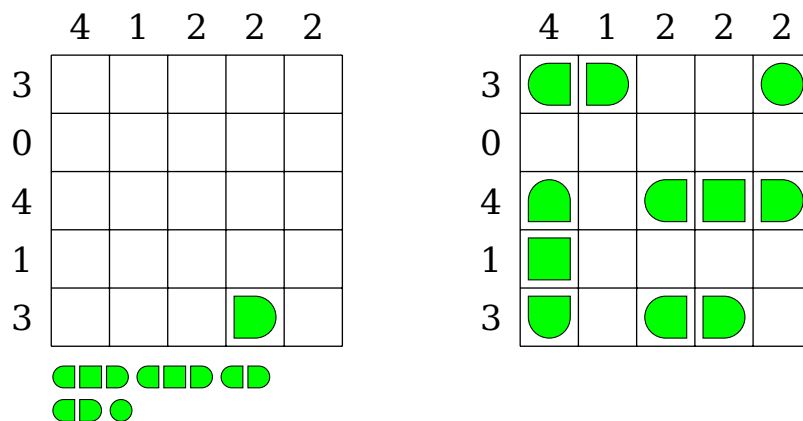
**cvoffset** [-23pt] sets the vertical offset of the counters in the margin.

## 1.2 Battleship

Try to find the positions of the ships listed below the puzzle. The numbers on the side of the puzzle reveals how many ship segments can be found in the rows and columns. All remaining fields indicate 'water'. Consider the following rules: The ships are arranged horizontally and vertically. No ship touches another ship at any point, not even diagonally.



## 1.2.1 Example



```

1 \begin{center}
2   \begin{battleship}
3     \placesegment{4}{1}{\ShipR}
4     \shipH{4,1,2,2,2}
5     \shipV{3,1,4,0,3}
6     \shipbox{3,3,2,2,1}
7   \end{battleship}
8   \hspace{1.5cm}
9   \begin{battleship}
10    \placeship{V}{1}{1}{3}
11    \placeship{H}{1}{5}{2}
12    \placeship{H}{3}{1}{2}
13    \placeship{H}{3}{3}{3}
14    \placeship{H}{5}{5}{1}
15    \shipH{4,1,2,2,2}
16    \shipV{3,1,4,0,3}
17  \end{battleship}
18 \end{center}

```

## 1.2.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid.

**shipcolor** [green] sets the color of the ship segments.

**width** [6cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [**Large**] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X sizes are used. Possible values: tiny, scriptsize, footnotesize, small, normalsize, large, Large, LARGE, huge, Huge

**title** [] sets the title of a puzzle.

**titleindent** [**0.75cm**] defines the indent of the title.

**titlewidth** [**5.15cm**] specifies the width of the box the title is set in.

**sbindent** [**0.75cm**] defines the indent of the ship box below the grid.

**sbwidth** [**5.15cm**] specifies the width of the minipage, in which the ships are typeset.

**sbshipscale** [**1**] scales the size of the ships in the ship box.

**bghcolor** [] sets the background color of the grid.

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

**cvoffset** [**-23pt**] sets the vertical offset of the counters in the margin.

## 1.3 Bokkusu

Black out some of the grid cells. The numbers on the left and the bottom edge of the grid indicate the values of the cells for adding up. The numbers on the right and the top edge of the grid specify the sums of the values of the colored cells.

### 1.3.1 Example

		7	1	11	9	6	
5							?
4							13
3							5
2							12
1							2
	1	2	3	4	5		

		7	1	11	9	6	
5							?
4							13
3							5
2							12
1							2
	1	2	3	4	5		

```

1 \begin{center}
2   \begin{bokkusu}
3     \valueH{1,2,3,4,5}

```

```

4      \valueV{1,2,3,4,5}
5      \sumH{7,1,11,9,6}
6      \sumV{2,12,5,13,?}
7      \end{bokkusu}
8      \hspace{1.5cm}
9      \begin{bokkusu}
10     \valueH{1,2,3,4,5}
11     \valueV{1,2,3,4,5}
12     \sumH{7,1,11,9,6}
13     \sumV{2,12,5,13,?}
14     \fillrow{5}{0,0,1,0,0}
15     \fillrow{4}{1,0,1,1,1}
16     \fillrow{3}{1,0,0,1,0}
17     \fillrow{2}{0,0,1,1,1}
18     \fillrow{1}{0,1,0,0,0}
19     \end{bokkusu}
20 \end{center}

```

### 1.3.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [6.7cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid.  
Here, the usual L<sup>A</sup>T<sub>E</sub>X sizes are used. Possible values: tiny, scriptsize, footnotesize, small, normalsize, large, Large, LARGE, huge, Huge

**title** [] sets the title of a puzzle.

**titleindent** [0.75cm] defines the indent of the title.

**titlewidth** [5.85cm] specifies the width of the box the title is set in.

**color** [black] specifies the color for coloring the cells.

**bgcolor** [] sets the background color of the grid.

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

**cvoffset** [-38pt] sets the vertical offset of the counters in the margin.

## 1.4 Chaos Sudoku

Fill the cells of an area with numbers from 1 to N of the N\*N grid. Each number can appear only once - in each area, column, row or diagonal if indicated.

### 1.4.1 Example

4				2
		4	5	
3				

4	3	5	1	2
2	1	3	4	5
5	4	2	3	1
1	2	4	5	3
3	5	1	2	4

```

1 \begin{center}
2   \begin{chaossudoku}
3     \chaossudokucell{1}{1}{3}
4     \chaossudokucell{1}{5}{4}
5     \chaossudokucell{3}{2}{4}
6     \chaossudokucell{4}{2}{5}
7     \chaossudokucell{5}{5}{2}
8     \begin{puzzlebackground}
9       \fillarea{Wheat}{(1,1)--(1,2)--(2,2)--(2,3)--(4,3)--(4,1)
10        --(1,1)}
11       \fillarea{HotPink!30}{(1,2)--(1,6)--(3,6)--(3,5)--(2,5)
12        --(2,2)--(1,2)}
13       \fillarea{GreenYellow}{(2,3)--(2,5)--(3,5)--(3,4)--(5,4)
14        --(5,2)--(4,2)--(4,3)--(2,3)}
15       \fillarea{LightBlue}{(3,4)--(3,6)--(6,6)--(6,5)--(5,5)
16        --(5,4)--(3,4)}
17       \fillarea{LightYellow}{(4,1)--(4,2)--(5,2)--(5,5)--(6,5)
18        --(6,1)--(4,1)}
19     \end{puzzlebackground}
20   \end{chaossudoku}
21   \hspace{1.5cm}
22   \begin{chaossudoku}
23     \setrow{5}{4,3,5,1,2}
24     \setrow{4}{2,1,3,4,5}
25     \setrow{3}{5,4,2,3,1}
26     \setrow{2}{1,2,4,5,3}
27     \setrow{1}{3,5,1,2,4}

```

```

28 \begin{puzzlebackground}
29   \fillarea{Wheat}{(1,1)--(1,2)--(2,2)--(2,3)--(4,3)--(4,1)
30     --(1,1)}
31   \fillarea{HotPink!30}{(1,2)--(1,6)--(3,6)--(3,5)--(2,5)
32     --(2,2)--(1,2)}
33   \fillarea{GreenYellow}{(2,3)--(2,5)--(3,5)--(3,4)--(5,4)
34     --(5,2)--(4,2)--(4,3)--(2,3)}
35   \fillarea{LightBlue}{(3,4)--(3,6)--(6,6)--(6,5)--(5,5)
36     --(5,4)--(3,4)}
37   \fillarea{LightYellow}{(4,1)--(4,2)--(5,2)--(5,5)--(6,5)
38     --(6,1)--(4,1)}
39 \end{puzzlebackground}
40 \end{chaossudoku}
41 \end{center}

```

### 1.4.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.1cm] 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** [-23pt] sets the vertical offset of the counters in the margin.

## 1.5 Hakyuu

Fill the cells of an area with numbers from 1 to SIZE of the area. If there are two cells with the same number N in a row or a column, there must be at least N cells between those two cells.

## 1.5.1 Example

2		6	5	
			4	
3				
	2			5
			1	

2	3	6	5	4
1	7	3	4	2
3	1	2	1	3
1	2	1	3	5
2	3	4	1	2

```

1 \begin{center}
2   \begin{hakyuu}
3     \hakyuucell{1}{5}{2}
4     \hakyuucell{3}{5}{6}
5     \hakyuucell{4}{5}{5}
6     \hakyuucell{4}{4}{4}
7     \hakyuucell{1}{3}{3}
8     \hakyuucell{2}{2}{2}
9     \hakyuucell{5}{2}{5}
10    \hakyuucell{4}{1}{1}
11    \begin{puzzlebackground}
12      \fillarea{Wheat}{(1,1)--(1,4)--(2,4)--(2,1)--(1,1)}
13      \fillarea{HotPink!30}{(1,4)--(1,6)--(6,6)--(6,5)--(3,5)
14        --(3,4)--(1,4)}
15      \fillarea{GreenYellow}{(2,4)--(3,4)--(3,5)--(5,5)--(5,4)
16        --(4,4)--(4,3)--(2,3)--(2,4)}
17      \fillarea{LightBlue}{(5,5)--(6,5)--(6,3)--(4,3)--(4,4)
18        --(5,4)--(5,5)}
19      \fillarea{LightSalmon!50}{(2,2)--(2,3)--(5,3)--(5,2)
20        --(2,2)}
21      \fillarea{LightYellow}{(2,1)--(2,2)--(5,2)--(5,3)--(6,3)
22        --(6,1)--(2,1)}
23    \end{puzzlebackground}
24  \end{hakyuu}
25  \hspace{1.5cm}
26  \begin{hakyuu}
27    \setrow{5}{2,3,6,5,4}
28    \setrow{4}{1,7,3,4,2}
29    \setrow{3}{3,1,2,1,3}
30    \setrow{2}{1,2,1,3,5}
31    \setrow{1}{2,3,4,1,2}
32    \begin{puzzlebackground}
33      \fillarea{Wheat}{(1,1)--(1,4)--(2,4)--(2,1)--(1,1)}

```

```

34 \fillarea{HotPink!30}{(1,4)--(1,6)--(6,6)--(6,5)--(3,5)
35 --(3,4)--(1,4)}
36 \fillarea{GreenYellow}{(2,4)--(3,4)--(3,5)--(5,5)--(5,4)
37 --(4,4)--(4,3)--(2,3)--(2,4)}
38 \fillarea{LightBlue}{(5,5)--(6,5)--(6,3)--(4,3)--(4,4)
39 --(5,4)--(5,5)}
40 \fillarea{LightSalmon!50}{(2,2)--(2,3)--(5,3)--(5,2)
41 --(2,2)}
42 \fillarea{LightYellow}{(2,1)--(2,2)--(5,2)--(5,3)--(6,3)
43 --(6,1)--(2,1)}
44 \end{puzzlebackground}
45 \end{hakyuu}
46 \end{center}

```

### 1.5.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.1cm] 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** [-23pt] sets the vertical offset of the counters in the margin.

## 1.6 Hitori

Black out some cells according to these specifications: In each row and each column a number may only occur once or can be completely blackened. The blackened cells can touch neither horizontal nor vertical. All non blackened cells must remain connected. Each number has its own color, which otherwise has no meaning.

## 1.6.1 Example

2	4	2	1	1
1	3	2	4	1
1	3	3	3	2
4	2	1	3	3
4	1	2	2	3

2	4		1	
	3	2	4	1
1		3		2
4	2	1	3	
	1		2	3

```

1 \begin{center}
2   \begin{hitori}
3     \framepuzzle
4     \setcolorrow{5}{2,4,2,1,1}
5     \setcolorrow{4}{1,3,2,4,1}
6     \setcolorrow{3}{1,3,3,3,2}
7     \setcolorrow{2}{4,2,1,3,3}
8     \setcolorrow{1}{4,1,2,2,3}
9   \end{hitori}
10  \hspace{1.5cm}
11  \begin{hitori}
12    \framepuzzle
13    \setcolorrow{5}{2,4,0,1,0}
14    \setcolorrow{4}{0,3,2,4,1}
15    \setcolorrow{3}{1,0,3,0,2}
16    \setcolorrow{2}{4,2,1,3,0}
17    \setcolorrow{1}{0,1,0,2,3}
18  \end{hitori}
19 \end{center}

```

## 1.6.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.1cm] 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** [-23pt] sets the vertical offset of the counters in the margin.

## 1.7 Kakuro

Enter numbers from 1 to 9 in any order into the blank cells. Here, the given horizontal and vertical sums should result. The zero does not occur. Within a summation, no number can be repeated.

### 1.7.1 Example

	23	16	10	
14				3
16				
14				
	8			

	23	16	10	
14	9	1	4	3
16	6	5	3	2
14	8	3	2	1
	8	7	1	

```

1 \definecolor{kakuro}{RGB}{155,206,167}
2 \kakurosetup{color=kakuro}
3 \begin{center}
4   \begin{kakuro}
5     \framepuzzle
6     \kakurorow{5}{\Black,\KKR{23}{},\KKR{16}{},\KKR{10}{},\Black}
7     \kakurorow{4}{\KKR{}{14},9,1,4,\KKR{}{3}}
8     \kakurorow{3}{\KKR{}{16},6,5,3,2}
9     \kakurorow{2}{\KKR{}{14},8,3,2,1}
10    \kakurorow{1}{\Black,\KKR{}{8},7,1,\Black}
11  \end{kakuro}
12  \hspace{1.5cm}
13  \begin{kakuro}[solution]
14    \framepuzzle

```

```

15 \kakurorow{5}{\Black,\KKR{23}{},\KKR{16}{},\KKR{10}{},\Black}
16 \kakurorow{4}{\KKR{14}{},9,1,4,\KKR{3}{}}
17 \kakurorow{3}{\KKR{16}{},6,5,3,2}
18 \kakurorow{2}{\KKR{14}{},8,3,2,1}
19 \kakurorow{1}{\Black,\KKR{8}{},7,1,\Black}
20 \end{kakuro}
21 \end{center}

```

### 1.7.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.1cm] specifies the width of the box the title is set in.

**color** [green] specifies the color of the kakuro cells.

**bgcolor** [] sets the background color of the grid.

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

**cvmoffset** [-23pt] sets the vertical offset of the counters in the margin.

**solution** [false] You can use the solution also for the puzzle, as the numbers in the cells are only typeset with option solution=true.

## 1.8 Kendoku

Fill the cells with the numbers from 1 to SIZE of the puzzle. In the top left corner of a framed area, you will find the result of the specified arithmetic function, which is applied on the entered numbers. The numbers may occur only once in each row and column. The numbers of an area may not necessarily be different when they are in different rows or columns.

## 1.8.1 Example

<sup>4+</sup>	<sup>2÷</sup>	<sup>75×</sup>		<sup>2</sup>
			<sup>2×</sup>	
<sup>5</sup>	<sup>60×</sup>			<sup>1</sup>
<sup>8×</sup>		<sup>2-</sup>	<sup>1-</sup>	
			<sup>8+</sup>	

<sup>4+</sup>	<sup>2÷</sup>	<sup>75×</sup>		<sup>2</sup>
1	4	3	5	2
3	2	5	<sup>2×</sup>	4
<sup>5</sup>	<sup>60×</sup>		<sup>1</sup>	1
<sup>8×</sup>		<sup>2-</sup>	<sup>1-</sup>	3
4	1	2	<sup>8+</sup>	5

```

1 \begin{center}
2   \begin{kendoku}
3     \framearea{black}{\tikzpath{1}{1}{8,8,6,2,6,2,4,4}}
4     \framearea{black}{\tikzpath{1}{3}{8,6,2,4}}
5     \framearea{black}{\tikzpath{1}{4}{8,8,6,2,2,4}}
6     \framearea{black}{\tikzpath{2}{2}{8,8,6,6,2,4,2,4}}
7     \framearea{black}{\tikzpath{2}{4}{8,8,6,2,2,4}}
8     \framearea{black}{\tikzpath{3}{1}{8,8,6,2,2,4}}
9     \framearea{black}{\tikzpath{3}{4}{8,8,6,6,2,4,2,4}}
10    \framearea{black}{\tikzpath{4}{1}{8,6,6,2,4,4}}
11    \framearea{black}{\tikzpath{4}{2}{8,6,6,2,4,4}}
12    \framearea{black}{\tikzpath{4}{3}{8,8,6,2,2,4}}
13    \framearea{black}{\tikzpath{5}{3}{8,6,2,4}}
14    \framearea{black}{\tikzpath{5}{4}{8,6,2,4}}
15    \framearea{black}{\tikzpath{5}{5}{8,6,2,4}}
16    \setrule{1}{2}{8\times}
17    \setrule{1}{3}{5}
18    \setrule{1}{5}{4+}
19    \setrule{2}{3}{60\times}
20    \setrule{2}{5}{2\div}
21    \setrule{3}{2}{2-}
22    \setrule{3}{5}{75\times}
23    \setrule{4}{1}{8+}
24    \setrule{4}{2}{1-}
25    \setrule{4}{4}{2\times}
26    \setrule{5}{3}{1}
27    \setrule{5}{5}{2}
28  \end{kendoku}
29  \hspace{1.5cm}
30  \begin{kendoku}
31    \framearea{black}{\tikzpath{1}{1}{8,8,6,2,6,2,4,4}}
32    \framearea{black}{\tikzpath{1}{3}{8,6,2,4}}
33    \framearea{black}{\tikzpath{1}{4}{8,8,6,2,2,4}}

```

```

34 \framearea{black}{\tikzpath{2}{2}{8,8,6,6,2,4,2,4}}
35 \framearea{black}{\tikzpath{2}{4}{8,8,6,2,2,4}}
36 \framearea{black}{\tikzpath{3}{1}{8,8,6,2,2,4}}
37 \framearea{black}{\tikzpath{3}{4}{8,8,6,6,2,4,2,4}}
38 \framearea{black}{\tikzpath{4}{1}{8,6,6,2,4,4}}
39 \framearea{black}{\tikzpath{4}{2}{8,6,6,2,4,4}}
40 \framearea{black}{\tikzpath{4}{3}{8,8,6,2,2,4}}
41 \framearea{black}{\tikzpath{5}{3}{8,6,2,4}}
42 \framearea{black}{\tikzpath{5}{4}{8,6,2,4}}
43 \framearea{black}{\tikzpath{5}{5}{8,6,2,4}}
44 \setrule{1}{2}{8\times}
45 \setrule{1}{3}{5}
46 \setrule{1}{5}{4+}
47 \setrule{2}{3}{60\times}
48 \setrule{2}{5}{2\div}
49 \setrule{3}{2}{2-}
50 \setrule{3}{5}{75\times}
51 \setrule{4}{1}{8+}
52 \setrule{4}{2}{1-}
53 \setrule{4}{4}{2\times}
54 \setrule{5}{3}{1}
55 \setrule{5}{5}{2}
56 \setrow{5}{1,4,3,5,2}
57 \setrow{4}{3,2,5,1,4}
58 \setrow{3}{5,3,4,2,1}
59 \setrow{2}{2,5,1,4,3}
60 \setrow{1}{4,1,2,3,5}
61 \end{kendoku}
62 \end{center}

```

### 1.8.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid.  
Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.1cm] 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** [-23pt] sets the vertical offset of the counters in the margin.

## 1.9 Killer Sudoku

Fill the cells with the numbers from 1 to SIZE of the puzzle. The numbers may occur only once in each row, column and colored area if specified. In the top left corner of a framed area, you will find the sum of the entered numbers. The numbers of an area may not necessarily be different, when they are in different rows or columns. But they must be different, when additional colored areas are specified.

### 1.9.1 Example

7	6	5	
			6
7			
	9		

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

```

1 \begin{killersudoku}
2   \framearea{black}{\tikzpath{1}{1}{8,8,6,6,2,4,2,4}}
3   \framearea{black}{\tikzpath{1}{3}{8,8,6,2,2,4}}
4   \framearea{black}{\tikzpath{2}{1}{8,6,6,6,2,4,4,4}}
5   \framearea{black}{\tikzpath{2}{3}{8,8,6,2,6,2,4,4}}
6   \framearea{black}{\tikzpath{3}{2}{8,6,8,6,2,2,4,4}}
7   \framearea{black}{\tikzpath{3}{4}{8,6,6,2,4,4}}
8   \begin{puzzlebackground}
9     \colorarea{orange!20}{\tikzpath{1}{1}{8,8,6,6,2,2,4,4}}
10    \colorarea{orange!20}{\tikzpath{3}{3}{8,8,6,6,2,2,4,4}}
11  \end{puzzlebackground}
12  \setrule{1}{2}{7}
13  \setrule{1}{4}{7}
14  \setrule{2}{1}{9}

```

```

15     \setrule{2}{4}{6}
16     \setrule{3}{4}{5}
17     \setrule{4}{3}{6}
18 \end{killersudoku}
19 \hspace{1.5cm}
20 \begin{killersudoku}
21     \framearea{black}{\tikzpath{1}{1}{8,8,6,6,2,4,2,4}}
22     \framearea{black}{\tikzpath{1}{3}{8,8,6,2,2,4}}
23     \framearea{black}{\tikzpath{2}{1}{8,6,6,6,2,4,4,4}}
24     \framearea{black}{\tikzpath{2}{3}{8,8,6,2,6,2,4,4}}
25     \framearea{black}{\tikzpath{3}{2}{8,6,8,6,2,2,4,4}}
26     \framearea{black}{\tikzpath{3}{4}{8,6,6,2,4,4}}
27 \begin{puzzlebackground}
28     \colorarea{orange!20}{\tikzpath{1}{1}{8,8,6,6,2,2,4,4}}
29     \colorarea{orange!20}{\tikzpath{3}{3}{8,8,6,6,2,2,4,4}}
30 \end{puzzlebackground}
31 \setrule{1}{2}{7}
32 \setrule{1}{4}{7}
33 \setrule{2}{1}{9}
34 \setrule{2}{4}{6}
35 \setrule{3}{4}{5}
36 \setrule{4}{3}{6}
37 \setrow{4}{3,2,4,1}
38 \setrow{3}{4,1,3,2}
39 \setrow{2}{2,4,1,3}
40 \setrow{1}{1,3,2,4}
41 \end{killersudoku}

```

### 1.9.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.1cm] specifies the width of the box the title is set in.

**bgcolor** [] sets the background color of the grid.

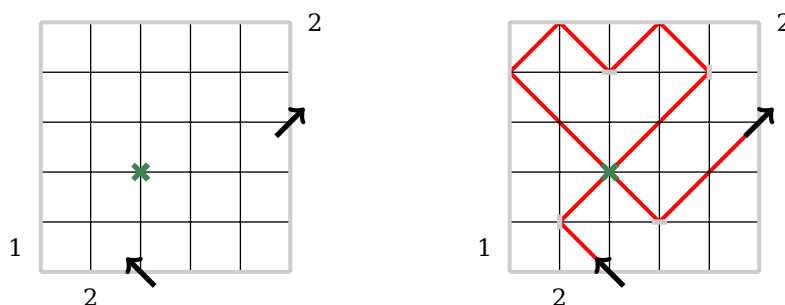
**counterstyle** [none] defines the counter style. Predefined styles: none, left, right

**cvoffset** [-23pt] sets the vertical offset of the counters in the margin.

## 1.10 Laser Beam

Draw a laser beam in each grid according to the following guidelines. The beam has to enter or to leave the grid at the arrows. At each intersection, a mirror, on which the laser beam must reflect on one side, can be placed horizontally or vertically. The other side must not be touched by the beam. All locations where the laser crosses are given. The numbers to the left and above the grid indicate how many cells are traversed by the beam in the corresponding row or column. The numbers to the right and below reveal, how many mirrors are found in the intersection of the corresponding row or column.

### 1.10.1 Example



```

1 \begin{center}
2   \begin{laserbeam}
3     \laserV{1}
4     \laserH{{}}
5     \mirrorH{{},2}
6     \mirrorV{{},{},{},{},{},2}
7     \framepuzzle[LP@cc@mirror]
8     \placearrow{3}{1}{LeftUp}
9     \placearrow{6}{4}{RightUp}
10    \placecross{3}{3}
11  \end{laserbeam}
12  \hspace{1cm}
13  \begin{laserbeam}
14    \laserV{1}

```

```

15 \laserH{{}}
16 \mirrorH{{},2}
17 \mirrorV{{},{},{},{},{},2}
18 \framepuzzle[LP@cc@mirror]
19 \placearrow{3}{1}{LeftUp}
20 \placearrow{6}{4}{RightUp}
21 \placecross{3}{3}
22 \placemirror{2}{2}{V}
23 \placemirror{4}{2}{H}
24 \placemirror{5}{5}{V}
25 \placemirror{3}{5}{H}
26 \begin{puzzlebackground}
27   \laser{\tikzpath{3}{1}{7,9,9,9,7,1,7,1,3,3,3,9,9}}
28 \end{puzzlebackground}
29 \end{laserbeam}
30 \end{center}

```

### 1.10.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [6.5cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [6.5cm] 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** [-38pt] sets the vertical offset of the counters in the margin.

## 1.11 Minesweeper








Draw a mine in some cells of the grid. The number in a cell indicates how many of the eight neighboring cells contain a mine. A numbered cell does not



contain a mine.

### 1.11.1 Example

	1			
		3	3	
3		4	2	
				0
	2			

	1			
		3	3	
3		4	2	
				0
	2			

```

1 \begin{center}
2   \begin{minesweeper}
3     \framepuzzle
4     \setrow{5}{{}},1}
5     \setrow{4}{{}},{ },3,3}
6     \setrow{3}{3,{ },4,2}
7     \setrow{2}{{}},{ },{ },{ },0}
8     \setrow{1}{{}},2}
9   \end{minesweeper}
10  \hspace{1.5cm}
11  \begin{minesweeper}
12    \framepuzzle
13    \setrow{5}{{}},1,{ },\Mine,\Mine}
14    \setrow{4}{{}},\Mine,3,3,\Mine}
15    \setrow{3}{3,\Mine,4,2}
16    \setrow{2}{{}},\Mine,\Mine,{ },0}
17    \setrow{1}{{}},2}
18  \end{minesweeper}
19 \end{center}

```

### 1.11.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [**Large**] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [**5.1cm**] specifies the width of the box the title is set in.

**bcolor** [] sets the background color of the grid.

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








**cvoffset** [**-23pt**] sets the vertical offset of the counters in the margin.

## 1.12 Schatzsuche

It's a variant of Minesweeper, just with diamonds instead of mines. Draw a diamond in some cells of the grid. The number in a cell indicates how many of the eight neighboring cells contain a diamond. A numbered cell does not contain a diamond.

### 1.12.1 Example

	1			
		3	3	
3		4	2	
				0
	2			

	1			
		3	3	
3		4	2	
				0
	2			

```

1 \begin{center}
2   \begin{schatzsuche}
3     \framepuzzle
4     \setrow{5}{{}},1
5     \setrow{4}{{}},{ },3,3
6     \setrow{3}{3,{ },4,2}
7     \setrow{2}{{}},{ },{ },{ },0
8     \setrow{1}{{}},2
9   \end{schatzsuche}

```

```

10 \hspace{1.5cm}
11 \begin{schatzsuche}
12   \framepuzzle
13   \setrow{5}{{}},1,{},\Diamond,\Diamond}
14   \setrow{4}{{}},\Diamond,3,3,\Diamond}
15   \setrow{3}{{}},\Diamond,4,2}
16   \setrow{2}{{}},\Diamond,\Diamond,{},0}
17   \setrow{1}{{}},2}
18 \end{schatzsuche}
19 \end{center}

```

### 1.12.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.1cm] 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** [-23pt] sets the vertical offset of the counters in the margin.

## 1.13 Skyline

There are skyscrapers located in each cell. Try to find out the height of the skyscraper in the respective cell. There are heights of 1 to MAX in every row, every column, and in each of the two diagonals if indicated. Some cells may be empty (parks). The numbers around the grid indicate how many buildings you can see from this position when you look at the skyscraper lineup. Bear in mind that only those skyscrapers are visible which are higher than the ones in front.

### 1.13.1 Example

		2		3	
3	2				3
4				3	1
		3		3	1

		2		3	
5	4	3	1	2	
4	5	1	2	3	
3	2	3	5	4	1
4	1	2	4	3	5
3	1	2	5	4	
		3		3	1

```

1 \begin{center}
2   \begin{skyline}
3     \skylineB{3, {}, 3, 1, {}}
4     \skylineL{{}, 4, 3, {}, {}}
5     \skylineT{{}, {}, 2, {}, 3}
6     \skylineR{{}, 1, 3, {}, {}}
7     \skylinecell{1}{3}{2}
8     \skylinecell{4}{2}{3}
9   \end{skyline}
10  \hspace{1cm}
11  \begin{skyline}
12    \skylineB{3, {}, 3, 1, {}}
13    \skylineL{{}, 4, 3, {}, {}}
14    \skylineT{{}, {}, 2, {}, 3}
15    \skylineR{{}, 1, 3, {}, {}}
16    \setrow{5}{5, 4, 3, 1, 2}
17    \setrow{4}{4, 5, 1, 2, 3}
18    \setrow{3}{2, 3, 5, 4, 1}
19    \setrow{2}{1, 2, 4, 3, 5}
20    \setrow{1}{3, 1, 2, 5, 4}
21  \end{skyline}
22 \end{center}

```

#### 1.13.1.1 Variants

##### 1.13.1.1.1 Skyline Sudoku

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

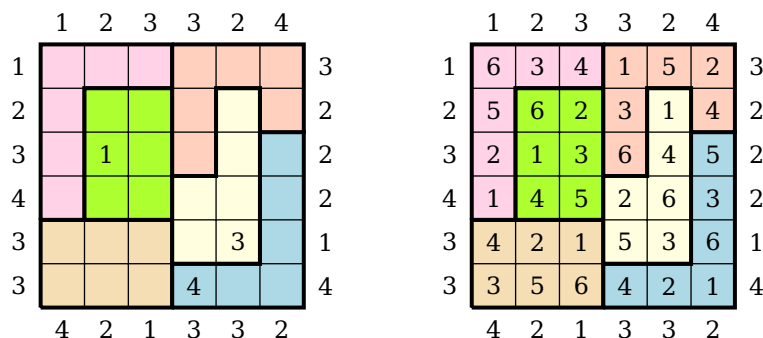
	4	1	3	2	3	5	3	2	3	
2	3	9	6	8	5	1	2	7	4	4
3	1	7	4	9	2	6	3	5	8	2
3	5	2	8	7	3	4	9	6	1	3
3	7	4	3	1	8	2	6	9	5	2
1	9	8	2	6	4	5	7	1	3	4
2	6	5	1	3	9	7	4	8	2	3
2	8	6	7	2	1	3	5	4	9	1
2	4	3	9	5	6	8	1	2	7	3
4	2	1	5	4	7	9	8	3	6	3
	4	5	2	5	2	1	2	4	3	

```

1 \begin{center}
2   \begin{skyline}[sudoku,scale=.4]
3     \skylineB{4,5,2,5,2,1,2,4,3}
4     \skylineL{4,2,2,2,1,3,3,3,2}
5     \skylineT{4,1,3,2,3,5,3,2,3}
6     \skylineR{3,3,1,3,4,2,3,2,4}
7     \setrow{9}{{},{},{},8,{{},{},{},7}}
8     \setrow{8}{{},{},{},4,{{},{},{},6,{{},{},{},8}}
9     \setrow{7}{{},{},{},2,{{},{},7,{{},{},{},{{},{},1}}
10    \setrow{6}{{},{},{},{},{{},{},8,2}}
11    \setrow{5}{{},{},{},{},2,{{},{},4,{{},{},7}}
12    \setrow{4}{{},{},{},{},{},3,{{},{},{{},{},4}}
13    \setrow{3}{{},{},{},{},{},{},1}}
14    \setrow{2}{{},{},{},3,{{},{},{},{{},{},1,2}}
15    \setrow{1}{{},{},{},{},5}}
16  \end{skyline}
17  \hspace{1cm}
18  \begin{skyline}[sudoku,scale=.4]
19    \skylineB{4,5,2,5,2,1,2,4,3}
20    \skylineL{4,2,2,2,1,3,3,3,2}
21    \skylineT{4,1,3,2,3,5,3,2,3}
22    \skylineR{3,3,1,3,4,2,3,2,4}
23    \setrow{9}{{3,9,6,8,5,1,2,7,4}}
24    \setrow{8}{{1,7,4,9,2,6,3,5,8}}
25    \setrow{7}{{5,2,8,7,3,4,9,6,1}}
26    \setrow{6}{{7,4,3,1,8,2,6,9,5}}
27    \setrow{5}{{9,8,2,6,4,5,7,1,3}}
28    \setrow{4}{{6,5,1,3,9,7,4,8,2}}
29    \setrow{3}{{8,6,7,2,1,3,5,4,9}}
30    \setrow{2}{{4,3,9,5,6,8,1,2,7}}
31    \setrow{1}{{2,1,5,4,7,9,8,3,6}}
32  \end{skyline}
33 \end{center}

```

## 1.13.1.1.2 Skyline Sudoku (N\*N)



```

1 \begin{center}
2   \begin{skyline}[rows=6,columns=6,scale=.58]
3     \skylineB{4,2,1,3,3,2}
4     \skylineL{3,3,4,3,2,1}
5     \skylineT{1,2,3,3,2,4}
6     \skylineR{4,1,2,2,2,3}
7     \skylinecell{2}{4}{1}
8     \skylinecell{4}{1}{4}
9     \skylinecell{5}{2}{3}
10    \begin{puzzlebackground}
11      \fillarea{Wheat}{(1,1)--(1,3)--(4,3)--(4,1)--(1,1)}
12      \fillarea{HotPink!30}{(1,3)--(1,7)--(4,7)--(4,6)--(2,6)
13        --(2,3)--(1,3)}
14      \fillarea{GreenYellow}{(2,3)--(2,6)--(4,6)--(4,3)--(2,3)}
15      \fillarea{LightBlue}{(4,1)--(7,1)--(7,5)--(6,5)--(6,2)
16        --(4,2)--(4,1)}
17      \fillarea{LightSalmon!50}{(4,7)--(4,4)--(5,4)--(5,6)--(6,6)
18        --(6,5)--(7,5)--(7,7)--(4,7)}
19      \fillarea{LightYellow}{(4,2)--(4,4)--(5,4)--(5,6)--(6,6)
20        --(6,2)--(4,2)}
21    \end{puzzlebackground}
22  \end{skyline}
23  \hspace{1cm}
24  \begin{skyline}[rows=6,columns=6,scale=.58]
25    \skylineB{4,2,1,3,3,2}
26    \skylineL{3,3,4,3,2,1}
27    \skylineT{1,2,3,3,2,4}
28    \skylineR{4,1,2,2,2,3}
29    \setrow{6}{6,3,4,1,5,2}
30    \setrow{5}{5,6,2,3,1,4}
31    \setrow{4}{2,1,3,6,4,5}
32    \setrow{3}{1,4,5,2,6,3}
33    \setrow{2}{4,2,1,5,3,6}

```

```

34 \setrow{1}{3,5,6,4,2,1}
35 \begin{puzzlebackground}
36   \fillarea{Wheat}{(1,1)--(1,3)--(4,3)--(4,1)--(1,1)}
37   \fillarea{HotPink!30}{(1,3)--(1,7)--(4,7)--(4,6)--(2,6)
38                       --(2,3)--(1,3)}
39   \fillarea{GreenYellow}{(2,3)--(2,6)--(4,6)--(4,3)--(2,3)}
40   \fillarea{LightBlue}{(4,1)--(7,1)--(7,5)--(6,5)--(6,2)
41                       --(4,2)--(4,1)}
42   \fillarea{LightSalmon!50}{(4,7)--(4,4)--(5,4)--(5,6)--(6,6)
43                           --(6,5)--(7,5)--(7,7)--(4,7)}
44   \fillarea{LightYellow}{(4,2)--(4,4)--(5,4)--(5,6)--(6,6)
45                           --(6,2)--(4,2)}
46 \end{puzzlebackground}
47 \end{skyline}
48 \end{center}

```

### 1.13.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**sudoku** [false] sets rows and columns to 9, in case of *⟨true⟩* is specified. Additionally the classic Sudoku grid is drawn.

**width** [6.7cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X sizes are used. Possible values: tiny, scriptsize, footnotesize, small, normalsize, large, Large, LARGE, huge, Huge

**title** [] sets the title of a puzzle.

**titleindent** [0.75cm] defines the indent of the title.

**titlewidth** [5.85cm] 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** [-38pt] sets the vertical offset of the counters in the margin.





### 1.14.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.2cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.2cm] 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** [-23pt] sets the vertical offset of the counters in the margin.

## 1.15 Sudoku

Well, it's Sudoku – nothing to explain! Fill each row and column with numbers from 1 to 9.

### 1.15.1 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

```

1 \begin{center}
2   \begin{lpudoku}
3     \setrow{9}{{}},2,6,{{}},{{}},{{}},{{}},{{}},{{}}
4     \setrow{8}{{}},{{}},{{}},{{}},{{}},{{}},1,7,{{}}
5     \setrow{7}{{}},{{}},3,1,{{}},6,{{}},{{}},{{}}
6     \setrow{6}{{}},6,{{}},{{}},5,{{}},8,{{}},3
7     \setrow{5}{{}},{{}},9,2,6,1,7,{{}},{{}}
8     \setrow{4}{5,{{}},4,{{}},8,{{}},{{}},6,{{}}
9     \setrow{3}{{}},{{}},{{}},8,{{}},4,3,{{}},{{}}
10    \setrow{2}{{}},4,8,{{}},{{}},{{}},{{}},{{}},{{}}
11    \setrow{1}{{}},{{}},{{}},{{}},{{}},{{}},{{}},9,4,{{}}
12  \end{lpudoku}
13  \hspace{1.5cm}
14  \begin{lpudoku}
15    \setrow{9}{1,2,6,5,7,8,4,3,9}
16    \setrow{8}{4,8,5,9,3,2,1,7,6}
17    \setrow{7}{7,9,3,1,4,6,5,8,2}
18    \setrow{6}{2,6,1,4,5,7,8,9,3}
19    \setrow{5}{8,3,9,2,6,1,7,5,4}
20    \setrow{4}{5,7,4,3,8,9,2,6,1}
21    \setrow{3}{6,5,2,8,9,4,3,1,7}
22    \setrow{2}{9,4,8,7,1,3,6,2,5}
23    \setrow{1}{3,1,7,6,2,5,9,4,8}
24  \end{lpudoku}

```

### 1.15.2 Options

**width** [**9.1cm**] sets the width of the minipage, in which the grid is typeset. 9 cells of width 1cm plus a little 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  $\frac{5}{9}$

**fontsize** [**Large**] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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.

**bicolor** [] sets the background color of the grid.

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

`cvoffset` [-23pt] sets the vertical offset of the counters in the margin.

### 1.15.3 Supporting bash scripts

#### 1.15.3.1 createlpsudoku

The `createlpsudoku` [2] bash script can transform Sudoku format files into `lpsudoku` environments. It can process files in the so called one line 81 format<sup>1</sup> (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

#### 1.15.3.2 lpsmag

With the `lpsmag` [23] bash script you can half automatically produce a Sudoku magazine using the  $\text{\LaTeX}$  package `lpsudoku.sty` and the `createlpsudoku` bash script.

Usage: `lpsmag configfile`

The script needs an installed `QQwing` [24] and a config file for defining the magazine's contents:

```

1 page p1 easy
2 page p2 easy
3 startpuzzles
4 typesetpage p1
5 typesetpage p2
6 startsolutions
7 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](#) [23] entry.

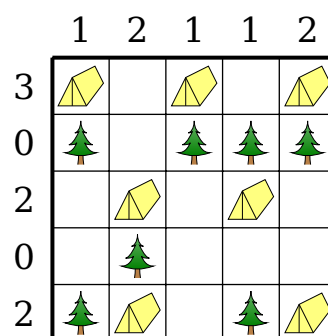
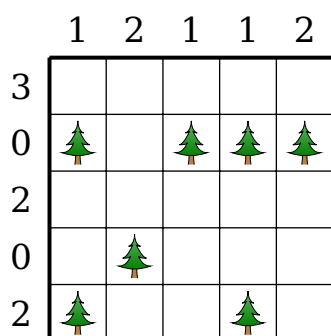
After running `lpsmag` you will find a `lpsmag.tex` in your working directory. Just run `pdflatex lpsmag.tex` twice and you finally get for example this [lpsmag.pdf](#).

<sup>1</sup>processing of several sudokus in 81 format (one in each line) is possible

## 1.16 Tents and Trees

Draw tents in the grid. Next to each tree, a tent must be entered in a horizontally or vertically adjacent cell, which is associated with this tree. The numbers next to the grid indicate the quantity of tents in each row or column. No tent can stand directly next to another one, not even diagonally.

### 1.16.1 Example



```

1 \begin{center}
2   \begin{tentsandtrees}
3     \framepuzzle
4     \tentH{1,2,1,1,2}
5     \tentV{2,0,2,0,3}
6     \setrow{4}{\Tree, {}, \Tree, \Tree, \Tree}
7     \setrow{2}{{}, \Tree}
8     \setrow{1}{\Tree, {}, {}, \Tree}
9   \end{tentsandtrees}
10  \hspace{1.5cm}
11  \begin{tentsandtrees}
12    \framepuzzle
13    \tentH{1,2,1,1,2}
14    \tentV{2,0,2,0,3}
15    \setrow{5}{\Tent, {}, \Tent, {}, \Tent,}
16    \setrow{4}{\Tree, {}, \Tree, \Tree, \Tree}
17    \setrow{3}{{}, \Tent, {}, \Tent}
18    \setrow{2}{{}, \Tree}
19    \setrow{1}{\Tree, \Tent, {}, \Tree, \Tent}
20  \end{tentsandtrees}
21 \end{center}

```

### 1.16.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.9cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.9cm] specifies the width of the box the title is set in.

**bgcolor** [] sets the background color of the grid.

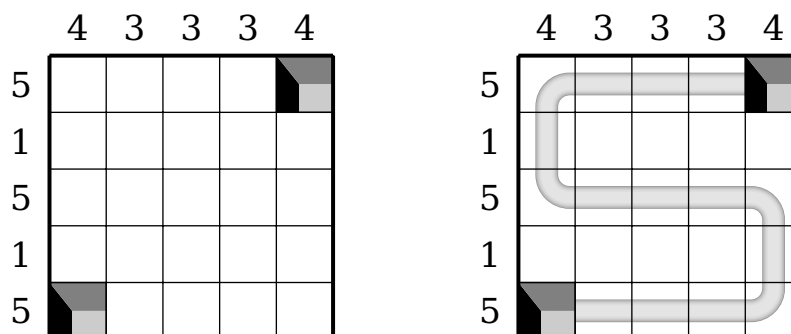
**counterstyle** [none] defines the counter style. Predefined styles: none, left, right

**cvoffset** [-23pt] sets the vertical offset of the counters in the margin.

### 1.17 Tunnel

Determine the course of the tube. Draw the only possible connection. from the beginning to the end. The numbers indicate how many tube segments (including portals) are present in the corresponding rows and columns. The tube is one cell wide, and does not cross or touch itself!

#### 1.17.1 Example



```

1 \begin{center}
2   \begin{tunnel}
3     \framepuzzle
4     \tunnelH{4,3,3,3,4}
5     \tunnelV{5,1,5,1,5}
6     \portal{1}{1}
7     \portal{5}{5}
8   \end{tunnel}
9   \hspace{1.5cm}
10  \begin{tunnel}
11    \framepuzzle
12    \tunnelH{4,3,3,3,4}
13    \tunnelV{5,1,5,1,5}
14    \portal{1}{1}
15    \portal{5}{5}
16    \tube{\tikzpath{1}{1}{6,6,6,6,8,8,4,4,4,4,8,6,6,6,6}}
17  \end{tunnel}
18 \end{center}

```

### 1.17.2 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.9cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [5.9cm] 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** [-23pt] sets the vertical offset of the counters in the margin.

## 2 Roll out your own grid-based logic puzzle

As an example we take a look at the former `bokkusu.sty` package. First, we ignore the LPPL license stuff.

```
\ProvidesPackage{bokkusu}[2013/03/25 bokkusu.sty v1.2 - Josef Kleber (C) 2013]%
\RequirePackage{logicpuzzle}%
```

We wrote a package `bokkusu.sty` with version number `v1.2` and date `2013/03/25` and added a copyright remark. We need to load the code base package `logicpuzzle.sty`.

```
\newcommand*\LP@BK@init@prefix{\LP@BK}%
\newcommand*\LP@BK@init@package{bokkusu}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{rows}{5}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{columns}{5}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{scale}{1}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{counterstyle}{none}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{color}{black}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{bgcolor}{}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{width}{6.7cm}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{cvmoffset}{-38pt}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{title}{}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{titleindent}{0.75cm}%
\LP@define@key{\LP@BK@init@prefix}{\LP@BK@init@package}{titlewidth}{5.85cm}%
\LP@define@choicekey@fontsize{\LP@BK@init@prefix}{\LP@BK@init@package}{Large}%
\ExecuteOptionsX{rows,columns,width,fontsize,scale,color,bgcolor,cvmoffset,
counterstyle,title,titleindent,titlewidth}%
\ProcessOptionsX\relax%
```

We save the package prefix and name in a macro for easy change. Then we define the options for package `bokkusu.sty` and the environment `bokkusu`, which are executed afterwards to create the macros for the option values.

```
\let\valueH\LP@bottomrow%
\let\valueV\LP@leftcolumn%
\let\sumH\LP@toprow%
\let\sumV\LP@rightcolumn%
```

We need numbers around the grid. Therefore, we define some aliases for the existing generic commands.

```
\newcommand*\bokkususetup[1]%
{%
  \setkeys{bokkusu.sty}{#1}%
}%
```

We define `\bokkususetup` for resetting the global package options.

Finally, we define the bokkusu environment.

```
\newenvironment{bokkusu}[1][1]{%
  {%
    \setkeys{bokkusu}{#1}%
    \LP@set@package{bokkusu}%
    \LP@set@env@prefix{LP@BK}%
    \setcounter{LP@rows}{\LP@BK@rows}%
    \setcounter{LP@columns}{\LP@BK@columns}%
    \stepcounter{LP@rows}%
    \stepcounter{LP@columns}%
  }
```

We locally set the environment options and the prefix and name of the current puzzle environment. We need to reset the counters for rows and columns, as they might have been altered.

```
\begin{minipage}[t]{\LP@BK@width}%
  \ifthenelse{\equal{\LP@BK@title}{}}{%
    {\par\enspace\par}% empty
  }{\enspace\par\noindent\hspace{\LP@BK@titleindent}\parbox{\LP@BK@titlewidth}
    {\strut\LP@titleformat\LP@BK@title}\vspace{3mm}\par}%
  \begin{tikzpicture}[scale=\LP@BK@scale]%
    \LP@drawbackground{1}{1}{\LP@BK@columns}{\LP@BK@rows}{\LP@BK@bgcolor}%
    \LP@drawgrid{1}{1}{\LP@BK@columns}{\LP@BK@rows}{1cm}%
  }\end{minipage}
```

We start a minipage with width  $\langle width \rangle$ . If the user defined a title, we typeset the title and add a vertical space. Then, we draw the puzzle with the help of tikz.sty. We start drawing the background and the grid.

```
{%
  \end{tikzpicture}%
  \LP@drawcounter{\LP@BK@counterstyle}%
  \stepcounter{LP@puzzlecounter}%
  \end{minipage}%
}%
```

Finally, we just end the picture for the puzzle. We draw and step the counter. As last action, we need to close the minipage environment. That's it. Easy, isn't it? You just need to copy this skelton and change or add some code for your specific puzzle.

## 3 The code

### 3.1 PGF layers

The logicpuzzle.sty package defines the PGF layers: LPdump, LPbgcolor, LPbackgroundtwo, LPbackground, LPforeground and LPforegroundtwo

Without specifying a special layer, the standard main layer is used. The LPbackground and LPforeground layers can be accessed with the puzzlebackground



[see: 3.2.2.1] and puzzleforeground [see: 3.2.2.2] environments. The LPbgcolor is and should only be used for the background color of the grid.

All layers can also be accessed with the generic PGF method:

```
\begin{pgfonlayer}{layer}
...
\end{pgfonlayer}
```

Order: LPdump → LPbgcolor → LPbackgroundtwo → LPbackground → main → LPforeground → LPforegroundtwo

So, if you are in the need to place something behind LPbackground or in front of LPforeground, you can use the LPbackgroundtwo and LPforegroundtwo layers. You can hide elements like help nodes behind the background color on the LPdump layer.

## 3.2 Environments

### 3.2.1 Puzzle environments

#### 3.2.1.1 logicpuzzle

```
\begin{logicpuzzle}{options}
...
\end{logicpuzzle}
```

The logicpuzzle environment is the generic environment for typesetting logic puzzles. With the optional argument of the environment, you can reset the options with local scope. Here, a blank grid is created. Furthermore, there are the puzzle environments described in section 1. They have their own set of options, that is also different option values and defaults! These can be changed with the \puzzlesetup commands with global scope or in the optional argument of the environment with local scope.

##### 3.2.1.1.1 Options

**rows** [5] defines the number of rows in the grid.

**columns** [5] specifies the number of columns in the grid

**width** [5.1cm] sets the width of the minipage, in which the grid is typeset.

**scale** [1] scales the size of the grid in the minipage.

**fontsize** [Large] specifies the size of the numbers next to the grid. Here, the usual L<sup>A</sup>T<sub>E</sub>X 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** [**5.1cm**] specifies the width of the box the title is set in.

**color** [] specifies the color for coloring the cells.

**bgcolor** [] sets the background color of the grid.

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

**cvoffset** [**-23pt**] sets the vertical offset of the counters in the margin.

### 3.2.2 Supporting environments

#### 3.2.2.1 puzzlebackground

`\begin{puzzlebackground}`  
`...`  
`\end{puzzlebackground}` The puzzlebackground environment allows you to place elements behind the main layer on the LPbackground layer [see: 3.1]. This is for example usefull for the `\fillarea` [see: 3.3.2.1.13] command.

#### 3.2.2.2 puzzleforeground

`\begin{puzzleforeground}`  
`...`  
`\end{puzzleforeground}` The puzzleforeground environment allows you to place elements in front of the main layer on the LPforeground layer [see: 3.1]. This is for example usefull for the `\framearea` [see: 3.3.2.1.12] command.

## 3.3 Commands

### 3.3.1 Puzzle specific commands

#### 3.3.1.1 2D-Sudoku







`\ddsudokucell{<column>}{<row>}`  
`{<number>}` **3.3.1.1.1 ddsudokucell** The command `\ddsudokucell` sets the *<number>* of the grid cell *<column>**<row>*.

`\ddsudokusetup{<options>}` **3.3.1.1.2 ddsudokusetup** With the command `\ddsudokusetup` you can reset the options with global scope.

#### 3.3.1.2 Battleship





`\placeship{<direction>}`  
`{<column>}{<row>}{<length>}` **3.3.1.2.1 placeship** With the command `\placeship` you can place complete ships in the grid. It expects the specification of the direction as horizontal (H) or vertical (V). Furthermore, it requires the starting coordinates and the length of the ship.

`\placesegment{⟨column⟩}{⟨row⟩}`  
`{⟨ship segment⟩}` **3.3.1.2.2 placesegment** The command `\placesegment` is used for the placement of ship segments in the grid. In the mandatory argument `⟨ship segment⟩`, you can use the following commands:

<code>\Ship</code>			<code>\ShipC</code>
<code>\ShipL</code>			<code>\ShipR</code>
<code>\ShipB</code>			<code>\ShipT</code>

**3.3.1.2.3 ship** The command `\ship` was replaced by the `\placesegment` command. The command `\ship` is deprecated and should not be used longer. It may still be used, but it is not recommended.

`\placewater{⟨column⟩}{⟨row⟩}` **3.3.1.2.4 placewater** With the command `\placewater` you can place water markers ( • ) in the grid.

`\placeisland{⟨column⟩}{⟨row⟩}` **3.3.1.2.5 placeisland** With the command `\placeisland` you can place islands (  ) in the grid. The island outlines are created randomly:    ...

`\shipH{⟨csv list⟩}` **3.3.1.2.6 shipH** The command `\shipH` typesets the horizontal numbers above the grid. It expects a comma-separated list as an argument.

`\shipV{⟨csv list⟩}` **3.3.1.2.7 shipV** The command `\shipV` typesets the vertical numbers beside the grid. It also expects a comma separated list.

`\shipbox{⟨csv list⟩}` **3.3.1.2.8 shipbox** The command `\shipbox` defines the number and size of the ships, which are typeset under the grid.

`\battleshipsetup{⟨options⟩}` **3.3.1.2.9 battleshipsetup** With the command `\battleshipsetup` you can reset the options with global scope.

`\classicgame{⟨csv list⟩}` **3.3.1.2.10 classicgame** The command `\classicgame` typesets a game sheet for playing classic Battleship. It expects a comma separated list with the number and sizes of the ships.

### 3.3.1.3 Bokkusu

`\valueH{⟨csv list⟩}` **3.3.1.3.1 valueH** The command `\valueH` typesets the numbers left to the grid indicating the values of the cells. It expects a comma-separated list as an argument.

`\valueV{<csv list>}` **3.3.1.3.2 valueV** The command `\valueV` typesets the numbers below the grid specifying the values of the cells. It also expects a comma separated list.

`\sumH{<csv list>}` **3.3.1.3.3 sumH** The command `\sumH` typesets the numbers right to the grid indicating the sums of the values of the colored cells. It expects a comma-separated list.

`\sumV{<csv list>}` **3.3.1.3.4 sumV** The command `\sumV` typesets the numbers above the grid specifying the sums of the values of the colored cells. It expects a comma separated list.

`\bokkususetup{<options>}` **3.3.1.3.5 bokkususetup** With the command `\bokkususetup` you can reset the options with global scope.

### 3.3.1.4 Chaos Sudoku

`\chaossudokucell{<column>}{<row>}{<number>}` **3.3.1.4.1 chaossudokucell** With the command `\chaossudokucell`, you can set the *<number>* of the grid cell *<column>* *<row>*.

`\chaossudokusetup{<options>}` **3.3.1.4.2 chaossudokusetup** With the command `\chaossudokusetup` you can reset the options with global scope.

### 3.3.1.5 Hakyuu

`\hakyuucell{<column>}{<row>}{<number>}` **3.3.1.5.1 hakyuucell** The command `\hakyuucell` sets the *<number>* of the grid cell *<column>* *<row>*.

`\hakyuusetup{<options>}` **3.3.1.5.2 hakyuusetup** With the command `\hakyuusetup` you can reset the options with global scope.

### 3.3.1.6 Hitori

`\hitorisetup{<options>}` **3.3.1.6.1 hitorisetup** With the command `\hitorisetup` you can reset the options with global scope.

### 3.3.1.7 Kakuro

`\kakurorow{<row>}{<csv list>}` **3.3.1.7.1 kakurorow** With the `\kakurorow` command, you can set the contents of a kakuro *<row>*. These may be numbers and the commands `\KKR` or `\Black`.

`\kakurocolumn{⟨column⟩}{⟨csv list⟩}` **3.3.1.7.2 kakurocolumn** With the `\kakurocolumn` command, you can set the contents of a kakuro `⟨column⟩`.

`\KKR{⟨sumV⟩}{⟨sumH⟩}` **3.3.1.7.3 KKR** With the `\KKR` command, you can set the contents of a kakuro cell.

`\Black` **3.3.1.7.4 Black** The command `\Black` blacks out a cell.

`\kakurosetup{⟨options⟩}` **3.3.1.7.5 kakurosetup** With the command `\kakurosetup` you can reset the options with global scope.

### 3.3.1.8 Kendoku

`\kendokucell{⟨column⟩}{⟨row⟩}{⟨number⟩}` **3.3.1.8.1 kendokucell** The command `\kendokucell` sets the `⟨number⟩` of the grid cell `⟨column⟩⟨row⟩`.

`\kendokusetup{⟨options⟩}` **3.3.1.8.2 kendokusetup** With the command `\kendokusetup` you can reset the options with global scope.

### 3.3.1.9 Killer Sudoku

`\killersudokucell{⟨column⟩}{⟨row⟩}{⟨number⟩}` **3.3.1.9.1 killersudokucell** The command `\killersudokucell` sets the `⟨number⟩` of the grid cell `⟨column⟩⟨row⟩`.

`\killersuduksetup{⟨options⟩}` **3.3.1.9.2 killersuduksetup** With the command `\killersuduksetup` you can reset the options with global scope.

### 3.3.1.10 Laser Beam

`\laserH{⟨csv list⟩}` **3.3.1.10.1 laserH** The command `\laserH` typesets the numbers above the grid indicating how many cells are traversed by the laser beam. It expects a comma-separated list as an argument.

`\laserV{⟨csv list⟩}` **3.3.1.10.2 laserV** The command `\laserV` typesets the numbers left to the grid.

`\mirrorH{⟨csv list⟩}` **3.3.1.10.3 mirrorH** The command `\mirrorH` typesets the numbers below the grid indicating how many mirrors are placed in the intersections of this column.

`\mirrorV{<csv list>}` **3.3.1.10.4 mirrorV** The command `\mirrorV` typesets the numbers right to the grid.

`\placearrow{<column>}{<row>}{<direction>}` **3.3.1.10.5 placearrow** The command `\placearrow` is used for the placement of arrows at the grid frame. The reference for coordinates is the bottom left corner of the cell. In the mandatory argument `<direction>`, you can use the following indicators: LeftUp, LeftDown, RightUp, RightDown

`\placecross{<column>}{<row>}` **3.3.1.10.6 placecross** With the command `\placecross` you can place a cross in the intersections of the grid.

`\placemirror{<column>}{<row>}{<direction>}` **3.3.1.10.7 placemirror** With the command `\placemirror` you can place mirrors in the intersections of the grid. In the mandatory argument `<direction>`, you can use the following indicators: H, V

`\laser{<color>}{<TikZ path>}` **3.3.1.10.8 laser** The command `\laser` draws the laser beam given by `<TikZ path>` with color `<color>` (default: red). The reference for coordinates is the bottom left corner of the cell.

```
\laser[green]{(1,2)--(2,3)--(1,4)}
```

You should consider using this command in the `puzzlebackground` environment.

`\laserbeamsetup{<options>}` **3.3.1.10.9 laserbeamsetup** With the command `\laserbeamsetup` you can reset the options with global scope.

### 3.3.1.11 Minesweeper

`\Mine` **3.3.1.11.1 Mine** The command `\Mine` draws a mine. It can be used in commands like `\setcell` [see: 3.3.2.1.1] or `\setrow` [see: 3.3.2.1.3]!

`\minesweeperssetup{<options>}` **3.3.1.11.2 minesweeperssetup** With the command `\minesweeperssetup` you can reset the options with global scope.

### 3.3.1.12 Schatzsuche

`\Diamond` **3.3.1.12.1 Diamond** The command `\Diamond` draws a diamond. It can be used in commands like `\setcell` [see: 3.3.2.1.1] or `\setrow` [see: 3.3.2.1.3]!

`\schatzsuchesetup{<options>}` **3.3.1.12.2 schatzsuchesetup** With the command `\schatzsuchesetup` you can reset the options with global scope.

### 3.3.1.13 Skyline

`\skylineT{⟨csv list⟩}` **3.3.1.13.1 skylineT** The command `\skylineT` typesets the numbers above the grid indicating how many skyscrapers are visible. It expects a comma-separated list as an argument.

`\skylineB{⟨csv list⟩}` **3.3.1.13.2 skylineB** The command `\skylineB` typesets the numbers below the grid.

`\skylineL{⟨csv list⟩}` **3.3.1.13.3 skylineL** The command `\skylineL` typesets the numbers left to the grid.

`\skylineR{⟨csv list⟩}` **3.3.1.13.4 skylineR** The command `\skylineR` typesets the numbers right to the grid.

`\skylinecell{⟨column⟩}{⟨row⟩}{⟨height⟩}` **3.3.1.13.5 skylinecell** The command `\skylinecell` sets the *⟨height⟩* of the grid cell *⟨column⟩⟨row⟩*.

`\skylinksetup{⟨options⟩}` **3.3.1.13.6 skylinksetup** With the command `\skylinksetup` you can reset the options with global scope.

### 3.3.1.14 Slitherlink

`\slitherlinkcell{⟨column⟩}{⟨row⟩}{⟨number⟩}` **3.3.1.14.1 slitherlinkcell** The command `\slitherlinkcell` sets the *⟨number⟩* of the grid cell *⟨column⟩⟨row⟩*.

`\slitherlinksetup{⟨options⟩}` **3.3.1.14.2 slitherlinksetup** With the command `\slitherlinksetup` you can reset the options with global scope.

### 3.3.1.15 Sudoku

`\lpsudokucell{⟨column⟩}{⟨row⟩}{⟨number⟩}` **3.3.1.15.1 lpsudokucell** The command `\lpsudokucell` sets the *⟨number⟩* of the grid cell *⟨column⟩⟨row⟩*.

`\lpsudokusetup{⟨options⟩}` **3.3.1.15.2 lpsudokusetup** With the command `\lpsudokusetup` you can reset the options with global scope.

### 3.3.1.16 Tents and Trees

`\tentH{⟨csv list⟩}` **3.3.1.16.1 tentH** The command `\tentH` typesets the numbers above the grid indicating how many tents are in the respective column. It expects a comma-separated list as an argument.

`\tentV{⟨csv list⟩}` **3.3.1.16.2 tentV** The command `\tentV` typesets the numbers left to the grid.

`\Tree` **3.3.1.16.3 Tree** The command `\Tree` draws a tree. It can be used in commands like `\setcell` [see: 3.3.2.1.1] or `\setrow` [see: 3.3.2.1.3]! The design of the tree is based on [Alain Matthes](#)’ answer to this [question](#) on T<sub>E</sub>X.sx.

`\Tent` **3.3.1.16.4 Tent** The command `\Tent` draws a tent. It can be used in commands like `\setcell` [see: 3.3.2.1.1] or `\setrow` [see: 3.3.2.1.3]!

`\tentsandtreessetup{⟨options⟩}` **3.3.1.16.5 tentsandtreessetup** With the command `\tentsandtreessetup` you can reset the options with global scope.

### 3.3.1.17 Tunnel

`\tunnelH{⟨csv list⟩}` **3.3.1.17.1 tunnelH** The command `\tunnelH` typesets the numbers above the grid indicating how many tube segments are in the respective column. It expects a comma-separated list as an argument.

`\tunnelV{⟨csv list⟩}` **3.3.1.17.2 tunnelV** The command `\tunnelV` typesets the numbers left to the grid.

`\portal{⟨column⟩}{⟨row⟩}` **3.3.1.17.3 portal** The command `\portal` is used for the placement of tunnel portals in the grid.

`\tube{⟨TikZ path⟩}` **3.3.1.17.4 tube** The command `\tube` draws the tunnel tube given by `⟨TikZ path⟩`. The reference for coordinates is the center of the cell. The design of the tube is based on [Xoff](#)’s answer to this [question](#) on T<sub>E</sub>X.sx.

```
\tube{(1.5,2.5)--(3.5,2.5)--(3.5,4.5)}
```

`\tunnelsetup{⟨options⟩}` **3.3.1.17.5 tunnelsetup** With the command `\tunnelsetup` you can reset the options with global scope.

## 3.3.2 User commands

### 3.3.2.1 In the grid

`\setcell{⟨⟨column⟩⟩}{⟨row⟩}{⟨element⟩}` **3.3.2.1.1 setcell** With the `\setcell` command, you can set `⟨element⟩` into cell `⟨column⟩⟨row⟩` as central node. It is aware of the current values of the



surrounding environment options rows, columns, scale and fontsize. Furthermore, a check if  $\langle element \rangle$  is within the grid is applied.

`\setbigcell[ $\langle fontsize \rangle$ ]{ $\langle column \rangle$ }{ $\langle row \rangle$ }{ $\langle element \rangle$ }` **3.3.2.1.2 setbigcell** The `\setbigcell` command sets  $\langle element \rangle$  into a big  $(2 \times 2)$  cell  $\langle column \rangle \langle row \rangle$  as central node. The optional argument  $\langle fontsize \rangle$  is set to 'Huge' by default.

`\setrow{ $\langle row \rangle$ }{ $\langle csv list \rangle$ }` **3.3.2.1.3 setrow** With the `\setrow` command, you can set the contents of a  $\langle row \rangle$ . These may be numbers or letters.

`\setcolorrow{ $\langle row \rangle$ }{ $\langle csv list \rangle$ }` **3.3.2.1.4 setcolorrow** With the `\setcolorrow` command, you can set the contents of a  $\langle row \rangle$ . Furthermore, the background of the cell is filled with color LP@{romannumber [see: 3.3.3.3.6]. With the number 0, you can black out the grid cell.

`\setcolumn{ $\langle column \rangle$ }{ $\langle csv list \rangle$ }` **3.3.2.1.5 setcolumn** With the `\setcolumn` command, you can set the contents of a  $\langle column \rangle$ . These may be numbers or letters.

`\setcolorcolumn{ $\langle column \rangle$ }{ $\langle csv list \rangle$ }` **3.3.2.1.6 setcolorcolumn** With the `\setcolorcolumn` command, you can set the contents of a  $\langle column \rangle$ . Furthermore, the background of the cell is filled with color LP@{romannumber [see: 3.3.3.3.6].

`\setrule{ $\langle column \rangle$ }{ $\langle row \rangle$ }{ $\langle rule \rangle$ }` **3.3.2.1.7 setrule** With the `\setrule` command, you can set a calculation rule  $\langle rule \rangle$  into the top left corner of cell  $\langle column \rangle \langle row \rangle$ . The rule is typeset in inline math mode. You might consider using the `\times` and `\div` commands.

`\fillcell{ $\langle column \rangle$ }{ $\langle row \rangle$ }` **3.3.2.1.8 fillcell** With the `\fillcell` command, you can fill cell  $\langle column \rangle \langle row \rangle$  with the color defined with environment option color<sup>2</sup>. It is aware of the current values of the surrounding environment options rows, columns, scale and color. Furthermore, a check if the cell is within the grid is applied.

`\fillrow{ $\langle row \rangle$ }{ $\langle csv list \rangle$ }` **3.3.2.1.9 fillrow** With the `\fillrow` command, you can fill a  $\langle row \rangle$ . In  $\langle csv list \rangle$  '1' means 'fill' and '0' means 'don't fill'. Internally, `\fillrow` uses `\fillcell` [see: 3.3.2.1.8].

`\fillcolumn{ $\langle column \rangle$ }{ $\langle csv list \rangle$ }` **3.3.2.1.10 fillcolumn** With the `\fillcolumn` command, you can fill a  $\langle column \rangle$ . In  $\langle csv list \rangle$  '1' means 'fill' and '0' means 'don't fill'. Internally, `\fillcolumn` uses `\fillcell` [see: 3.3.2.1.8].

<sup>2</sup>Therefore, you must define an option color in the style file you want to use fill commands

`\filldiagonals[color]` **3.3.2.1.11 filldiagonals** With the `\filldiagonals` command, you can fill the diagonals with the color specified with the optional argument *color* (default: yellow!20). Furthermore, it checks for a quadratic grid, otherwise an error message is issued.

`\framearea{color}{TikZ path}` **3.3.2.1.12 framearea** The command `\framearea` frames the area given by *TikZ path* with color *color*. The reference for coordinates is the bottom left corner of the cell.

```
\framearea{green}{(2,2)--(2,3)--(3,3)--(3,2)--(2,2)}
```

This command will color the frame of the grid cell (2,2) green. You should consider using this command in the `puzzleforeground` [see: 3.2.2.2] environment.

`\fillarea{color}{TikZ path}` **3.3.2.1.13 fillarea** The command `\fillarea` fills the area given by *TikZ path* with color *color*. The reference for coordinates is the bottom left corner of the cell. You should consider using this command in the `puzzlebackground` [see: 3.2.2.1] environment.

`\colorarea{color}{TikZ path}` **3.3.2.1.14 colorarea** The command `\colorarea` fills the area given by *TikZ path* with color *color* – just like `\framearea` without frame.

`\framepuzzle[color]` **3.3.2.1.15 framepuzzle** With the `\framepuzzle` command, you can frame the grid (thicker line) with the color specified with the optional argument *color* (default: black).

`\tikzpath{column}{row}{csv list}` **3.3.2.1.16 tikzpath** With the `\tikzpath` command, you can easily construct a TikZ path. You just need to define a starting point *column**row* (bottom left corner) and a *csv list* with direction indicators relative to the current position.

7: up left	8: up	9: up right
4: left	5: no change	6: right
1: down left	2: down	3: down right

```
\framearea{green}{\tikzpath{2}{2}{8,6,2,4}}
```

This command will frame grid cell (2,2) green.

### 3.3.2.2 Presentation

`\titleformat{<format>}` **3.3.2.2.1 titleformat** With the `\titleformat` command, you can define the `<format>` of the title. By default, the definition is as follows:

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

`\puzzlecounter` **3.3.2.2.2 puzzlecounter** The command `\puzzlecounter` provides the general puzzle counter in textual form to use it in `\definecounterstyle`.

`\setpuzzlecounter{<number>}` **3.3.2.2.3 setpuzzlecounter** With the command `\setpuzzlecounter`, you can reset the puzzle counter, for example before the solutions.

`\definecounterstyle{<name>}`  
`{<definition>}` **3.3.2.2.4 definecounterstyle** 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@cvmoffset}\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`.

`\setgridlinestyle{<style>}` **3.3.2.2.5 setgridlinestyle** The command `\setgridlinestyle` sets the style of lines used in the grid. By default, the style is set to solid, whereas `slitherlink.sty` uses dashed.

`\setnormallinewidth{<dimension>}` **3.3.2.2.6 setnormallinewidth** With the command `\setnormallinewidth`, you can set the width of the standard lines (default: 0.5pt)

`\setthicklinewidth{<dimension>}` **3.3.2.2.7 setthicklinewidth** With the command `\setthicklinewidth`, you can set the width of the 'thicker' lines (default: 1.5pt)

### 3.3.3 Internal commands

#### 3.3.3.1 Initialization

```
\LP@define@key{<prefix>}{<package>}{<option>}{<default>}
```

**3.3.3.1.1 LP@define@key** With the `\LP@define@key` command, you can define the options of the environment `<package>`. A `<prefix>` is needed for creating different name spaces.

```
\LP@define@key{LP@BS}{battleship}{rows}{5}
```

This code snippet defines the option `rows` for environment `battleship` with the default value 5. This value is stored in `\LP@BS@rows`.

```
\LP@define@choicekey@fontsize{<prefix>}{<package>}{<default>}
```

**3.3.3.1.2 LP@define@choicekey@fontsize** With this command, you can define the choice key option `fontsize` of the environment `<package>`. Possible keys are: `tiny`, `scriptsize`, `footnotesize`, `small`, `normalsize`, `large`, `Large`, `LARGE`, `huge`, `Huge`

### 3.3.3.2 Drawing grids

```
\LP@drawgrid{<xmin>}{<ymin>}{<xmax>}{<ymax>}{<step>}
```

**3.3.3.2.1 LP@drawgrid** With the `\LP@drawgrid` command, you can draw the grid `((<xmin>,<ymin>))` to `((<xmax>,<ymax>))` with step `<step>`. For drawing the standard puzzle grid the step must be 1cm.

```
\LP@drawsudokugrid
```

**3.3.3.2.2 LP@drawsudokugrid** The command `\LP@drawsudokugrid` draws the standard Sudoku grid, but just the thicker lines. You will have to overlay the standard grid to get a full Sudoku grid.

```
\LP@drawbackground{<xmin>}{<ymin>}{<xmax>}{<ymax>}{<color>}
```

**3.3.3.2.3 LP@drawbackground** The command `\LP@drawbackground` draws the background color of the grid.

### 3.3.3.3 In the grid

```
\LP@LP@setcellcontent{<column>}{<row>}{<element>}
```

**3.3.3.3.1 LP@setcellcontent** The command `\LP@setcellcontent` is the generic command to set an arbitrary `<element>`.

```
\LP@LP@setcellcontentC{<column>}{<row>}{<element>}
```

**3.3.3.3.2 LP@setcellcontentC** The command `\LP@setcellcontentC` is the generic command to set an arbitrary `<element>` in a centered node in the bottom left corner.

```
\LP@setrowcontents{<csv list>}{<column>}{<row>}
```

**3.3.3.3.3 LP@setrowcontents** The command `\LP@setrowcontents` is the generic command to set row contents. It does not necessarily start with `<column> 1!`

```
\LP@setcolumncontents{<csv list>}{<column>}{<row>}
```

**3.3.3.3.4 LP@setcolumncontents** The command `\LP@setcolumncontents` is the generic command to set column contents. It does not necessarily start with `<row> 1!`

```
\LP@ingrid
{<column>}{<row>}{<max column>}
{<max row>}{<package>}
```

**3.3.3.3.5 LP@ingrid** With the `\LP@ingrid` command, you can check if an element – that should be placed – is within the grid. Otherwise an error message is issued.

```
\LP@definecolor
{<name>}{<rgb color>}
```

**3.3.3.3.6 LP@definecolor** With the `\LP@definecolor` command, you can define named rgb colors, especially for defining background colors of numbers used in `\setcolorrow` [see: 3.3.2.1.4] and `\setcolorcolumn` [see: 3.3.2.1.6].

The background color names follow the pattern: `LP@c@romannumber`

```
\LP@definecolor{LP@c@iv}{.55,1,.88}
```

This command will define the new background color of number **4**!

### 3.3.3.4 Around the grid

```
\LP@leftcolumn{<csv list>}
```

**3.3.3.4.1 LP@leftcolumn** With the `\LP@leftcolumn` command, you can set the contents of the column left to the grid. The `\skylineL` command for the skyline environment is for example defined as follows:

```
\let\skylineL\LP@leftcolumn
```

```
\LP@rightcolumn{<csv list>}
```

**3.3.3.4.2 LP@rightcolumn** With the `\LP@rightcolumn` command, you can set the contents of the column right to the grid.

```
\LP@toprow{<csv list>}
```

**3.3.3.4.3 LP@toprow** With the `\LP@toprow` command, you can set the contents of the row above the grid.

```
\LP@bottomrow{<csv list>}
```

**3.3.3.4.4 LP@bottomrow** With the `\LP@bottomrow` command, you can set the contents of the row below the grid.

### 3.3.3.5 Presentation

```
\LP@drawcounter{<name>}
```

**3.3.3.5.1 LP@drawcounter** The command `\LP@drawcounter` draws the puzzle counter with counter style `<name>`.



## 4 Examples

You can download application examples and their solutions from the [project page](#). The puzzles are originally licensed under .

## References

- [1] Bernhard Hobiger. HoDoKu, 2012.  
(<http://hodoku.sourceforge.net/en/index.php>).
- [2] Josef Kleber. createlopsudoku, 2013.  
(<http://logicpuzzle.square7.de/createlopsudoku>).
- [3] Josef Kleber. Examples & Solutions (battleship), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-battleship.pdf>).
- [4] Josef Kleber. Examples & Solutions (bokkusu), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-bokkusu.pdf>).
- [5] Josef Kleber. Examples & Solutions (chaossudoku), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-chaossudoku.pdf>).
- [6] Josef Kleber. Examples & Solutions (ddsudoku), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-ddsudoku.pdf>).
- [7] Josef Kleber. Examples & Solutions (hakyuu), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-hakyuu.pdf>).
- [8] Josef Kleber. Examples & Solutions (hitori), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-hitori.pdf>).
- [9] Josef Kleber. Examples & Solutions (kakuro), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-kakuro.pdf>).
- [10] Josef Kleber. Examples & Solutions (kendoku), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-kendoku.pdf>).
- [11] Josef Kleber. Examples & Solutions (killersudoku), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-killersudoku.pdf>).
- [12] Josef Kleber. Examples & Solutions (laserbeam), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-laserbeam.pdf>).
- [13] Josef Kleber. Examples & Solutions (lpsudoku), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-lpsudoku.pdf>).

- [14] Josef Kleber. Examples & Solutions (minesweeper), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-minesweeper.pdf>).
- [15] Josef Kleber. Examples & Solutions (schatzsuche), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-schatzsuche.pdf>).
- [16] Josef Kleber. Examples & Solutions (skyline), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-skyline.pdf>).
- [17] Josef Kleber. Examples & Solutions (slitherlink), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-slitherlink.pdf>).
- [18] Josef Kleber. Examples & Solutions (tentsandtrees), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-tentsandtrees.pdf>).
- [19] Josef Kleber. Examples & Solutions (tunnel), 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/downloads/example-tunnel.pdf>).
- [20] Josef Kleber. How to create a Sudoku magazine, 2013.  
([http://logicpuzzle.square7.de/sudoku\\_magazine](http://logicpuzzle.square7.de/sudoku_magazine)).
- [21] Josef Kleber. logicpuzzle – A LaTeX style file for typesetting logic puzzles, 2013. (<http://logicpuzzle.square7.de/>).
- [22] Josef Kleber. The logicpuzzle bundle, 2013.  
(<https://bitbucket.org/kleberj/logicpuzzle/>).
- [23] Josef Kleber. lpsmag, 2013. (<http://logicpuzzle.square7.de/lpsmag>).
- [24] Stephen Ostermiller. QQwing – Sudoku Generator and Solver, 2011.  
(<http://ostermiller.org/qqwing/>).

**B**

battleship environment .... 52  
 \battleshipsetup ..... 43  
 bgcolor environment option 8, 10,  
 11, 13, 15, 17, 18, 21, 23,  
 24, 26, 27, 31, 33, 34, 37,  
 38, 42  
 \Black ..... 44, 45  
 bokkusu environment .... 39, 40  
 bokkusu.sty ..... 39  
 \bokkususetup ..... 39, 44

**C**

\chaossudokucell ..... 44  
 \chaossudokusetup ..... 44  
 \classicgame ..... 43  
 <color> mandatory argument . 50,  
 52  
 <color> optional argument 46, 50  
 color environment option 11, 18,  
 42, 49  
 \colorarea ..... 50  
 <column> mandatory argument 42–  
 50, 52, 53  
 columns environment option 8, 9,  
 11, 13, 15, 16, 18, 20, 22,  
 24, 25, 27, 31, 33, 37, 38,  
 40, 41, 49  
 counterstyle environment option  
 . 8, 10, 11, 13, 15, 17, 18,  
 21, 23, 24, 26, 27, 31, 33,  
 34, 37, 38, 42  
 <csv list> mandatory argument 43–  
 50, 52, 53  
 cvoffset environment option . 8,  
 10, 11, 13, 15, 17, 18, 21,  
 23, 24, 26, 27, 31, 33, 35,  
 37, 38, 42, 51

**D**

\ddsudokucell ..... 42  
 \ddsudokusetup ..... 42  
 <default> mandatory argument 52  
 \definecounterstyle ..... 51  
 <definition> mandatory argument  
 ..... 51  
 \Diamond ..... 46

<dimension> mandatory argument  
 ..... 51  
 <direction> mandatory argument  
 ..... 42, 46  
 \div ..... 49

**E**

<element> mandatory argument 48,  
 49, 52  
 environment  
 battleship ..... 52  
 bokkusu ..... 39, 40  
 logicpuzzle ..... 41  
 lpsudoku ..... 35  
 minipage ..... 40  
 puzzlebackground 40, 42, 46,  
 50  
 puzzleforeground 41, 42, 50  
 skyline ..... 53  
 environment option  
 bgcolor 8, 10, 11, 13, 15, 17,  
 18, 21, 23, 24, 26, 27, 31,  
 33, 34, 37, 38, 42  
 color ..... 11, 18, 42, 49  
 columns 8, 9, 11, 13, 15, 16,  
 18, 20, 22, 24, 25, 27, 31,  
 33, 37, 38, 40, 41, 49  
 counterstyle . 8, 10, 11, 13,  
 15, 17, 18, 21, 23, 24, 26,  
 27, 31, 33, 34, 37, 38, 42  
 cvoffset 8, 10, 11, 13, 15, 17,  
 18, 21, 23, 24, 26, 27, 31,  
 33, 35, 37, 38, 42, 51  
 fontsize 8, 10, 11, 13, 15, 16,  
 18, 20, 22, 24, 26, 27, 31,  
 33, 34, 37, 38, 41, 49, 52  
 rows 8, 9, 11, 13, 15, 16, 18,  
 20, 22, 24, 25, 27, 31, 33,  
 37, 38, 40, 41, 49, 52  
 sbindent ..... 10  
 sbshipscale ..... 10  
 sbwidth ..... 10  
 scale 8, 9, 11, 13, 15, 16, 18,  
 20, 22, 24, 25, 27, 31, 33,  
 34, 37, 38, 41, 49  
 solution ..... 18  
 sudoku ..... 31



- title 8, 10, 11, 13, 15, 17, 18,  
20, 22, 24, 26, 27, 31, 33,  
34, 37, 38, 41
- titleindent 8, 10, 11, 13, 15,  
17, 18, 20, 22, 24, 26, 27,  
31, 33, 34, 37, 38, 41
- titewidth 8, 10, 11, 13, 15,  
17, 18, 21, 22, 24, 26, 27,  
31, 33, 34, 37, 38, 42
- width 8, 9, 11, 13, 15, 16, 18,  
20, 22, 24, 25, 27, 31, 33,  
34, 37, 38, 41
- F**
- \fillarea ..... 42, 50
- \fillcell ..... 49
- \fillcolumn ..... 49
- \filldiagonals ..... 50
- \fillrow ..... 49
- <fontsize> optional argument . 49
- fontsize environment option . 8,  
10, 11, 13, 15, 16, 18, 20,  
22, 24, 26, 27, 31, 33, 34,  
37, 38, 41, 49, 52
- <format> mandatory argument 51
- \framearea ..... 42, 50
- \framepuzzle ..... 50
- H**
- \hakyuucell ..... 44
- \hakyuusetup ..... 44
- <height> mandatory argument . 47
- \hitorisetup ..... 44
- K**
- \kakurocolumn ..... 45
- \kakurorow ..... 44
- \kakurosetup ..... 45
- \kendokucell ..... 45
- \kendokusetup ..... 45
- \killersudokucell ..... 45
- \killersudokusetup ..... 45
- \KKR ..... 44, 45
- L**
- \laser ..... 46
- \laserbeamsetup ..... 46
- \laserH ..... 45
- \laserV ..... 45
- <length> mandatory argument 42
- logicpuzzle environment .... 41
- logicpuzzle.sty ..... 39, 40
- \LP ..... 52, 53
- \LP@bottomrow ..... 53
- \LP@BS@rows ..... 52
- \LP@define@key ..... 52
- \LP@definecolor ..... 53
- \LP@drawbackground ..... 52
- \LP@drawgrid ..... 52
- \LP@drawsudokugrid ..... 52
- \LP@ingrid ..... 53
- \LP@leftcolumn ..... 53
- \LP@rightcolumn ..... 53
- \LP@setcellcontent ..... 52
- \LP@setcellcontentC ..... 52
- \LP@setcolumncontents .... 52
- \LP@setrowcontents ..... 52
- \LP@toprow ..... 53
- LPbackground PGF layer .. 40–42
- LPbackgroundtwo PGF layer . 40,  
41
- LPbgcolor PGF layer .... 40, 41
- LPdump PGF layer ..... 40, 41
- LPforeground PGF layer .. 40–42
- LPforegroundtwo PGF layer . 40,  
41
- lpsudoku environment ..... 35
- lpsudoku.sty ..... 35
- \lpsudokucell ..... 47
- \lpsudokusetup ..... 47
- M**
- main PGF layer ..... 40–42
- mandatory argument
  - <color> ..... 50, 52
  - <column> .... 42–50, 52, 53
  - <csv list> ..... 43–50, 52, 53
  - <default> ..... 52
  - <definition> ..... 51
  - <dimension> ..... 51
  - <direction> ..... 42, 46
  - <element> ..... 48, 49, 52
  - <format> ..... 51
  - <height> ..... 47
  - <length> ..... 42
  - <max column> ..... 53

- `<max row>` ..... 53
- `<name>` ..... 51, 53
- `<number>` . 42, 44, 45, 47, 51
- `<option>` ..... 52
- `<options>` ..... 42–48
- `<package>` ..... 52, 53
- `<prefix>` ..... 52
- `<rgb color>` ..... 53
- `<row>` ..... 42–50, 52, 53
- `<rule>` ..... 49
- `<ship segment>` ..... 43
- `<step>` ..... 52
- `<style>` ..... 51
- `<sumH>` ..... 45
- `<sumV>` ..... 45
- `<TikZ path>` ..... 46, 48, 50
- `<width>` ..... 40
- `<xmax>` ..... 52
- `<xmin>` ..... 52
- `<ymax>` ..... 52
- `<ymin>` ..... 52
- `\marginnote` ..... 51
- `<max column>` mandatory argument  
..... 53
- `<max row>` mandatory argument 53
- `\Mine` ..... 46
- `\minesweeperssetup` ..... 46
- minipage environment ..... 40
- `\mirrorH` ..... 45
- `\mirrorV` ..... 46
- N**
- `<name>` mandatory argument 51,  
53
- `<number>` mandatory argument 42,  
44, 45, 47, 51
- O**
- `<option>` mandatory argument 52
- optional argument
  - `<color>` ..... 46, 50
  - `<fontsize>` ..... 49
  - `<options>` ..... 41
- `<options>` mandatory argument 42–  
48
- `<options>` optional argument .. 41
- P**
- `<package>` mandatory argument 52,  
53
- PGF layer
  - `LPbackground` ..... 40–42
  - `LPbackgroundtwo` ... 40, 41
  - `LPbgcolor` ..... 40, 41
  - `LPdump` ..... 40, 41
  - `LPforeground` ..... 40–42
  - `LPforegroundtwo` ... 40, 41
  - main ..... 40–42
- `\placearrow` ..... 46
- `\placecross` ..... 46
- `\placeisland` ..... 43
- `\placemirror` ..... 46
- `\placesegment` ..... 43
- `\placeship` ..... 42
- `\placewater` ..... 43
- `\portal` ..... 48
- `<prefix>` mandatory argument 52
- puzzlebackground environment .  
..... 40, 42, 46, 50
- `\puzzlecounter` ..... 51
- puzzleforeground environment .  
..... 41, 42, 50
- R**
- `\reversemarginpar` ..... 51
- `<rgb color>` mandatory argument  
..... 53
- `<row>` mandatory argument 42–50,  
52, 53
- rows environment option 8, 9, 11,  
13, 15, 16, 18, 20, 22, 24,  
25, 27, 31, 33, 37, 38, 40,  
41, 49, 52
- `<rule>` mandatory argument .. 49
- S**
- sbindent environment option 10
- sbshipscale environment option  
..... 10
- sbwidth environment option . 10
- scale environment option 8, 9, 11,  
13, 15, 16, 18, 20, 22, 24,  
25, 27, 31, 33, 34, 37, 38,  
41, 49
- `\schatzsuchesetup` ..... 46

- `\setbigcell` ..... 49
- `\setcell` ..... 46, 48
- `\setcolorcolumn` ..... 49, 53
- `\setcolorrow` ..... 49, 53
- `\setcolumn` ..... 49
- `\setgridlinestyle` ..... 51
- `\setnormallinewidth` ..... 51
- `\setpuzzlecounter` ..... 51
- `\setrow` ..... 46, 48, 49
- `\setrule` ..... 49
- `\setthicklinewidth` ..... 51
- `\ship segment` mandatory argument  
..... 43
- `\shipbox` ..... 43
- `\shipH` ..... 43
- `\shipV` ..... 43
- skyline environment ..... 53
- `\skylineB` ..... 47
- `\skylinecell` ..... 47
- `\skylineL` ..... 47, 53
- `\skylineR` ..... 47
- `\skylinesetup` ..... 47
- `\skylineT` ..... 47
- `slitherlink.sty` ..... 51
- `\slitherlinkcell` ..... 47
- `\slitherlinksetup` ..... 47
- solution environment option 18
- `\step` mandatory argument . 52
- `\style` mandatory argument .. 51
- sudoku environment option ... 31
- `\sumH` ..... 44
- `\sumH` mandatory argument 45
- `\sumV` ..... 44
- `\sumV` mandatory argument 45
- T**
- `\Tent` ..... 48
- `\tentH` ..... 47
- `\tentsandtreessetup` ..... 48
- `\tentV` ..... 48
- `\TikZ path` mandatory argument  
..... 46, 48, 50
- `\tikz` ..... 51
- `tikz.sty` ..... 40
- `\tikzpath` ..... 50
- `\times` ..... 49
- title environment option . 8, 10,  
11, 13, 15, 17, 18, 20, 22,  
24, 26, 27, 31, 33, 34, 37,  
38, 41
- `\titleformat` ..... 51
- titleindent environment option  
. 8, 10, 11, 13, 15, 17, 18,  
20, 22, 24, 26, 27, 31, 33,  
34, 37, 38, 41
- titlewidth environment option 8,  
10, 11, 13, 15, 17, 18, 21,  
22, 24, 26, 27, 31, 33, 34,  
37, 38, 42
- `\Tree` ..... 48
- `\tube` ..... 48
- `\tunnelH` ..... 48
- `\tunnelsetup` ..... 48
- `\tunnelV` ..... 48
- V**
- `\valueH` ..... 43
- `\valueV` ..... 44
- W**
- `\width` mandatory argument 40
- width environment option 8, 9, 11,  
13, 15, 16, 18, 20, 22, 24,  
25, 27, 31, 33, 34, 37, 38,  
41
- X**
- `\xmax` mandatory argument . 52
- `\xmin` mandatory argument . 52
- Y**
- `\ymax` mandatory argument . 52
- `\ymin` mandatory argument . 52