

The *tikzrput* package ¹

Alain Matthes

March 8, 2016

¹<http://altermundus.com/pages/tkz/tikzrput/>

☞ (Version 0.2)

This document describes the L^AT_EX package *tikzrput*. It also provides some examples and comments on the package's use. Firstly, I would like to thank a lot of contributors of the site [tex.stackexchange](http://tex.stackexchange.com) ². The idea to create this package comes from a question on [tex.stackexchange](http://tex.stackexchange.com). I would like to thanks also **Till Tantau** for creating the wonderful tool **TikZ**.

²<http://tex.stackexchange.com/>

How to install the package

With TeXLive, if you need to install it by yourself, just download the file `tikzrput.sty`, and place it in your TDS directory (`/texmf/tex/latex` for Unix-like systems).

With MiKTeX, copy the file `tikzrput.sty` into `C:\texmf\tex\latex`, then run **MiKTeX Options** . In the **File name database** section, click on **Refresh now**.

How to use the package

You only need to add

`\usepackage{tikzrput}` in your preamble. The `tikzrput` package loads **TikZ**. If `pstricks` is already loaded, the macro is unchanged. The macro `\rput` of this package is only active with **pdf_latex**.

The macro

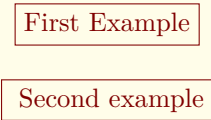
The `\rput` macro can be used to place objects. The simple form of the `\rput` macro used below works like the `\put` macro of the L^AT_EX `picture` environment and gives you the ability to place and rotate whatever you want.

Below you can see the common usage of this command. `\rput(x,y){text}` to print text at the ref point (x,y) . In the following example the usage of this command is shown practically.

```
\documentclass{article}
\usepackage{tikzrput}
\begin{document}
  ...
  see code below
  ...
\end{document}
```

baseline->

```
baseline->\rput(12,1){%
\begin{tikzpicture}
\node[draw]{First Example};
\end{tikzpicture}}
\rput(12,0){\fbox{%
Second example}}
```



"(x,y)" is the position where "stuff" will be placed."Refpoint" gives the reference point of stuff (text or picture and so on).

baseline->

```
baseline->\rput{45}(12,0){\fbox{Stuff}}
```



The specifications of the \rput command is:

```
\rput [<options>]{<rotation>}(refpoint){stuff}
```

The first mandatory argument (in parenthesis) is the coordinate pair of the point where the stuff is placed.

The second mandatory argument (in curly braces) is the stuff to place.

The first optional argument is given in brackets and it determines the position of the bounding box of the object to place with respect to the "refpoint" (in parenthesis). The admissible values are mc, t, tl, tr, b, bl, br and B, Br, Bl. The default is mc meaning middle – center, t is for top of the box, b for bottom and B for baseline; r and l are for right and left. We can call these points, the anchors.

The second optional argument is given in curly braces. It is a number that stands for the rotation angle.

What is the baseline ?

As you probably know T_EX puts all its objects in boxes. A box has a baseline that determines height and depth.

T_EX uses the baselines for fixing together the boxes to others. A box has ten anchors:

- mc the middle-center, the center point,
- bl br tl tr are the 4 corners (for bottom left and bottom right top left and top right),
- two anchors Bl and Br (Baseline left and right),
- the vertical line through the middle-center defines three other points t, B and b on the top line, the baseline and the bottom line.

\rput command will place the box defined by using the reference point and placing on this, one of the ten anchors box.

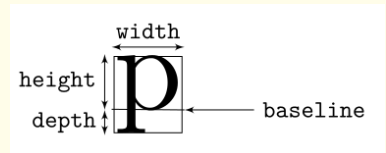


Figure 1: The baseline

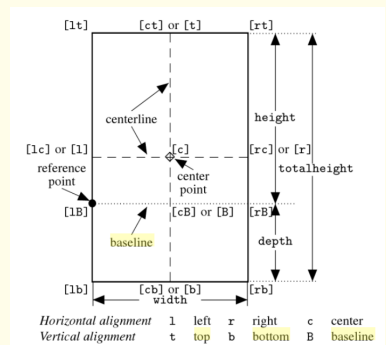


Figure 2: Anchors of a box

Arguments and options

Available anchors:

name	default	definition
<code>mc</code>	{ }	middle center mc or empty
<code>B</code>	{ }	baseline center
<code>Bl</code>	{ }	baseline left
<code>Br</code>	{ }	baseline right
<code>t</code>	{ }	top center
<code>tl</code>	{ }	top left
<code>tr</code>	{ }	top right
<code>b</code>	{ }	bottom center
<code>bl</code>	{ }	bottom left
<code>br</code>	{ }	bottom right
<code>rotation</code>	0	angle of rotation around ref point

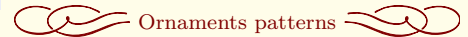
Table 1: List of options for the `pgfornament` macro.

The angle of *rotation* is expressed in degrees...

The *refpoint* is defined by two coordinates (x,y).

*Examples**ex 1 Ornaments patterns*

```
\hspace{1cm}%
\rlput[r]{-3pt,3pt}{\pgfornament[scale=.2]{72}}
  {Ornaments patterns}%
\rlput[l]{3pt,3pt}{\large\pgfornament[scale=.2]{73}}
```

*ex 2 Ornaments*

```
\documentclass{scrartcl}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[dvipsnames]{xcolor}
\usepackage{tikzrput}
\usepackage{pgfornament}
\begin{document}
\tikzset{pgfornamentstyle/.style={%
draw = Periwinkle,
  fill = SpringGreen}}
\unitlength=1cm
\begin{picture}(10,10)%
  \color{blue}%
  \put(0,0){\framebox(10,10){%
  \rlput[tl]{-3,5}{\pgfornament[width=6cm]{71}}}%
  \rlput[bl]{-3,-5}{\pgfornament[width=6cm,
  symmetry=h]{71}}}%
  \rlput[tl]{-5,5}{\pgfornament[width=2cm]{63}}}%
  \rlput[tr]{5,5}{\pgfornament[width=2cm,
  symmetry=v]{63}}}%
  \rlput[bl]{-5,-5}{\pgfornament[width=2cm,
  symmetry=h]{63}}}%
  \rlput[br]{5,-5}{\pgfornament[width=2cm,
  symmetry=c]{63}}}%
  \rlput[bl]{-90}{-5,3}{\pgfornament[width=6cm]{46}}}%
  \rlput[bl]{90}{5,-3}{\pgfornament[width=6cm]{46}}}%
  \rlput(0,0){\Huge \color{MidnightBlue} Ornaments}%
  \rlput[t]{0,-0.5}{\pgfornament[width=5cm]{75}}}%
  \rlput[b]{0,0.5}{\pgfornament[width=5cm]{69}}}%
  \rlput[tr]{-30}{-1,2.5}{\pgfornament[width=2cm]{57}}}%
  \rlput[tl]{30}{1,2.5}{\pgfornament[width=2cm,symmetry=v]{57}}}}%
\end{picture}
\end{document}
```



Figure 3: caption

ex3 Picture

```
\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[dvipsnames]{xcolor}
```

```

\usepackage{pict2e,tikzrput}
\usepackage[calc]{picture}
\usepackage[]{fourier}
\begin{document}

\setlength{\parindent}{0pt}
\setlength{\unitlength}{1cm}
\begin{picture}(3,3)
  \put(0,0){\framebox(3,3){}}%
  \color{blue}%
  \thicklines
  \put(0,0){\line(1,1){3}}
  \color{MidnightBlue}%
  \rput[b]{45}(2,2){\large \textbf{line}}
\end{picture}%

\end{document}

```

ex 4 Tikzpicture

```

\documentclass{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[dvipsnames]{xcolor}
\usepackage{pict2e,tikzrput}
\usepackage[calc]{picture}
\usepackage{fourier}
\begin{document}

\hrule
baseline%
\begin{tikzpicture}[baseline=(current bounding box.south west)]
  \draw[help lines] (0,0) grid (3,3) ;
  \draw[use as bounding box,color=CadetBlue] (0,0)rectangle(3,3);
  \rput (2,2){Perfect}%
\end{tikzpicture}%
baseline%

\end{document}

```

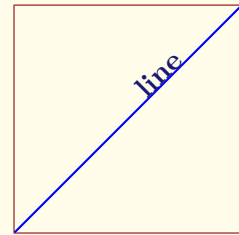


Figure 4: with picture

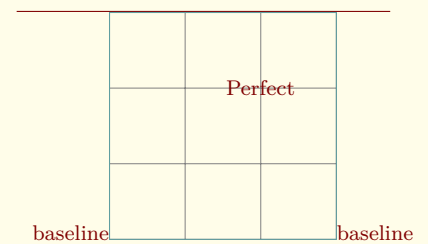


Figure 5: with tikzpicture

ex 5 From pstricks doc

As we have already seen, the `{\rput}` macro can be used to place objects. The second mandatory argument (in curly braces) is the stuff to place the first mandatory argument (in parenthesis) is the coordinate pair of the point where the stuff is placed. Now we turn to the optional arguments of the `{\rput}` macro. The first one is given in brackets. It determines the justification of the bounding box of the object to place with respect to the point given in parenthesis. The admissible values are the same as the values for the option origin of the `{\includegraphics}` macro. For an instance `{[br]}` for bottom-right. The default is mc meaning middle - center. The second optional argument is given in curly braces just before the left parenthesis. It is a number that stands for the rotation angle as illustrated in the last instance of the `{\rput}` macro on the slide. The two optional arguments make `{\rput}` more exible than the `{\put}` macro of the picture environment.

```
\rput[b](-2,0){%
\begin{tikzpicture}[%
every node/.style={inner sep=0pt}]
\tikzset{image/.style={circle,
fill=red,
minimum size = 4pt,
inner sep = 0pt,
outer sep = 1pt}
}
\node[inner sep = 1cm] (wrapper){\tikz
\node(image) {\pgfornament[scale=1.5,opacity=0.2]{1}};};
\end{tikzpicture}}
```

ex 6 From pstricks again

```
\documentclass{scrartcl}
\PassOptionsToPackage{dvipsnames,svgnames}{xcolor}
\usepackage{tikzrput}
\begin{document}

\rput{30}(7,-5){Tu as de beaux yeux !}
\rput(8,-2){\myEye}
\rput(6,-2){\reflectbox{\myEye}}
\end{document}
```

ex 7 Ancres

Utilisation des ancrs

```
\documentclass{scrartcl}
\PassOptionsToPackage{dvipsnames,svgnames}{xcolor}
\usepackage{tikzrput}
\begin{document}
```

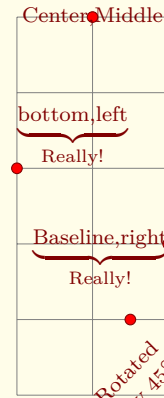
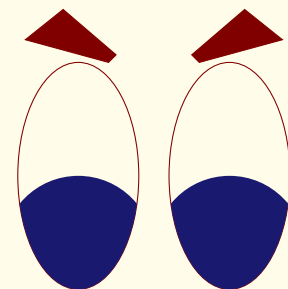


Figure 6: from pstricks

Figure 7: from pstricks again



Tu as de beaux yeux !

```
->\rput[] (5,0){\tikz \draw[blue] (0,0) rectangle +(1,2);}
\rput[tl] (5,0){\tikz \draw[red] (0,0) rectangle +(1,2);}
\rput[br] (5,0){\tikz \draw[green] (0,0) rectangle +(1,2);}
\rput[Bl] (5,0){\tikz \draw[orange](0,0) rectangle +(1,2);}
\end{document}
```

->

Figure 8: anrcs

