the ptltools package[*]

Peterlits Zo
peterlitszo@outlook.com

June 18, 2020

Contents

[*] This doument corresponds to ptltools v0.0.1, dated 2020/6/18

1 First at All

This document is for the package `ptltools`, which has a kit of tools to build great document you like: status bars, a to-do boxes, and so on.

If you like to add others and you are good at LaTeX-ing, it is a good idea that go to `www.github.com/peterlitszo/ptltools` to push your request pull. Or email me by your idea, then I'd like to try to achieve it as my best.

2 Source code and the more details

2.1 The Head of the File

This is the head of `ptltools`'s package:

```
\NeedsTeXFormat{LaTeX2e}[2005/12/01]
\ProvidesPackage{ptltools}
    [2020/6/18 v0.0.1 a kit of tools, greater than your thinking]
\RequirePackage{tikz}
\RequirePackage{ifthen}
```

2.2 The Todo Boxes

2.2.1 The Lower-Level Todo Boxes

There are four todo boxes for your: empty todo box, doing todo box, ok todobox and error todobox. the lower-level command is showed by the table 1.

`\usebox\ptl@todo@empty` □	`\usebox\ptl@todo@doing` ☒
`\usebox\ptl@todo@ok` ☑	`\usebox\ptl@todo@error` ⍰

Table 1: The pictures of raw to-do boxes

```
\newsavebox{\ptl@todo@empty}
\newsavebox{\ptl@todo@doing}
\newsavebox{\ptl@todo@ok}
\newsavebox{\ptl@todo@error}

\newlength{\p@b}
\setlength{\p@b}{0.9ex}

\newcommand{\ptl@todo@inital}{
    \savebox{\ptl@todo@empty}{%
        \tikz{
            \path[use as bounding box] (0, 0) rectangle (3\p@b, 2\p@b);
            \draw (0,0) rectangle (2\p@b,2\p@b);
        }%
    }
    \savebox{\ptl@todo@doing}{%
        \tikz{
            \path[use as bounding box] (0, 0) rectangle (3\p@b, 2\p@b);
            \draw (1.5\p@b,2\p@b) -- (0,2\p@b) -- (0,0) -- (2\p@b,0)
                  -- (2\p@b,0.5\p@b);
            \draw (0.75\p@b,1.25\p@b) -- (1.5\p@b,0.5\p@b) -- (3\p@b,2\p@b);
            \draw (1.75\p@b,1.75\p@b) -- (2.75\p@b,0.75\p@b);
        }%
    }
    \savebox{\ptl@todo@ok}{%
        \tikz{
            \path[use as bounding box] (0, 0) rectangle (3\p@b, 2\p@b);
            \draw (2\p@b,1.5\p@b) -- (2\p@b,2\p@b) -- (0,2\p@b) -- (0,0)
                  -- (2\p@b,0) -- (2\p@b,0.5\p@b);
            \draw (0.75\p@b,1.25\p@b) -- (1.5\p@b,0.5\p@b) -- (3\p@b,2\p@b);
        }%
    }
    \savebox{\ptl@todo@error}{%
        \tikz{
            \path[use as bounding box] (0, 0) rectangle (3\p@b, 2\p@b);
            \draw (0,0) rectangle (2\p@b,2\p@b);
            \draw (1\p@b, 0.5\p@b) -- (1\p@b,0.75\p@b)
                  .. controls (1\p@b,1\p@b) and (1.5\p@b,1\p@b) ..
                      (1.5\p@b,1.2\p@b)
                  .. controls (1.5\p@b,1.75\p@b) and (1.25\p@b,1.75\p@b) ..
                      (1\p@b,1.75\p@b)
                  .. controls (0.75\p@b,1.75\p@b) and (0.5\p@b,1.75\p@b) ..
                      (0.5\p@b,1.2\p@b);
            \fill (1\p@b,0.25\p@b) circle [radius=0.1\p@b];
        }%
    }
}

\ptl@todo@inital
```

2.2.2 The Higher-Level Boxes

It is a good idea to use commands like \ptltodo[ ] but not the lower-level commands. Because those commands are more readable than these lower-level commands.

Look at table 2 for more infomations.

```
\def\ptltodo[#1]{%
    \ifthenelse{\equal{#1}{\space}}{%
        \usebox\ptl@todo@empty%
    }{\ifthenelse{\equal{#1}{x}}{%
        \usebox\ptl@todo@doing%
    }{\ifthenelse{\equal{#1}{v}}{%
```

`\ptltodo[ ]` □	`\ptltodo[x]` ☒
`\ptltodo[v]` ☑	`\ptltodo[others]` ⍰

Table 2: The pictures of to-do boxes for user

```
        \usebox\ptl@todo@ok%
    }{%
        \usebox\ptl@todo@error%
    }}}%
}
```