the `ptltools` package *

Peterlits Zo†

June 21, 2020

Contents

*This doument corresponds to `ptltools` v0.0.1, dated 2020/6/21

†Email: `< peterlitszo@outlook.com >`

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}
    [\the\year/\the\month/\the\day{}
     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

`\ptl@todo@empty` `\ptl@todo@doing` `\ptl@todo@ok` `\ptl@todo@error`

There are four todo boxes for you: 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

Those commands declare four base raw todo boxes:

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

`\ptl@todo@baselength` Those boxes are based on the length – `\ptl@todo@baselength`, and all the boxes have the size 2`\ptl@todo@baselength`× 3`\ptl@todo@baselength`:

```
\newlength{\ptl@todo@baselength}
\setlength{\ptl@todo@baselength}{0.9ex}
```

`\ptl@todo@inital` The next part defines command `\ptl@todo@inital` to inital all boxes register by TikZ boxes.

```
\newcommand{\ptl@todo@inital}{
    \def\p@b{\ptl@todo@baselength}
```

1. (In the definition of `\ptl@todo@inital`) This will make box `\ptl@todo@empty` have the value of "☐ ".

```
    \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);
        }%
    }
```

2. (In the definition of `\ptl@todo@inital`) This will make box `\ptl@todo@doing` have the value of "☒".

```
    \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);
        }%
    }
```

3. (In the definition of `\ptl@todo@inital`) This will make box `\ptl@todo@ok` have the value of "☒".

```
    \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);
        }%
    }
```

4. (In the definition of `\ptl@todo@inital`) This will make box `\pt@todo@error` have the value of "☐ ".

```
    \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];
        }%
    }
```

End of its definition.

```
}
\ptl@todo@inital
```

2.2.2 The Higher-Level Boxes

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

Look at table 2 for more infomations.

`\ptltodo[ ]`	`\ptltodo[x]`
`\ptltodo[v]`	`\ptltodo[others]`

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

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

Change History

v0.0.1

General: Initial version 1

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.