

The `todonotes` package*

Henrik Skov Midtiby
`henrikmidtiby@gmail.com`

June 4, 2021

Abstract

The `todonotes` package allows you to insert to-do items in your document. At any point in the document a list of all the inserted to-do items can be listed with the `\listoftodos` command.

Contents

1	Introduction	2
1.1	Usage	2
1.2	Package options	4
1.3	Options for the <code>todo</code> command	5
1.4	Options for the <code>missingfigure</code> command	8
1.5	Options for the <code>listoftodos</code> command	9
1.6	Known issues	10
1.7	Things to improve	13
1.8	Usage methods	13
2	Implementation	20
2.1	Declaration of options for the package	20
2.2	Options for the <code>todo</code> command	24
2.3	The main code part	27

*This document corresponds to `todonotes` v1.1.5, dated 2021/06/04.

1 Introduction

The `todonotes` package makes three commands available to the user: `\todo[]{}`, `\missingfigure{}` and `\listoftodos`. `\todo[]{}` and `\missingfigure{}` makes it possible to insert notes in your document about things that has to be done later (`todonotes . . .`). I developed the basic functionality of the package while I worked on my bachelor project.

Some alternatives for the `todonotes` package are:

- [easy-todo](#)
Depends on `color`, `tocloft` and `ifthen`, small feature set.
- [fixmetodonotes](#)
Depends on `graphicx`, `color`, `transparent`, `watermark`, `fix-cm`, `ulem` and `tocloft`, small feature set.
- [todo](#)
Depends on `amssymb`, medium feature set.
- [fixme](#)
Large package with a lot of features.

The main reason for considering other packages is that the `todonotes` package is quite large and relies heavily on `tikz`. This can slow down compilation of large documents significantly. The mentioned alternatives have a different feature set and does not rely on `tikz`, which makes them require fewer resources.

1.1 Usage

`\todo` My most common usage of the `todonotes` package, is to insert an `todonotes` somewhere in a latex document. An example of this usage is the command

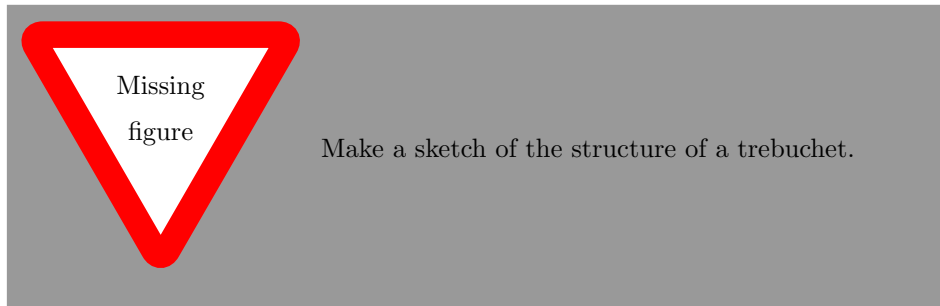
```
\todo{Make a cake \ldots},
```

which renders like. The `\todo` command has this structure: `\todo[options]{todo text}`. The `todo text` is the text that will be shown in the `todonote` and in the list of `todos`. The optional argument `options`, allows the user to customize the appearance of the inserted `todonotes`. For a description of all the options see section 1.3.

Make a cake . . .
























`\missingfigure` The `\missingfigure` command inserts an image containing an attention sign and the given text. The command takes only one argument `\missingfigure{text}`, a text string that could describe what the figure should consist of. An example of its usage could be
















```
\missingfigure{Make a sketch of the structure of a trebuchet.}
which renders like.
```



`\listoftodos` The `\listoftodos` command inserts a list of all the todos in the current document. `\listoftodos` takes no arguments. For this document the list of to-do's looks like.

Todo list

 Make a cake ...	2
Figure: Make a sketch of the structure of a trebuchet.	2
 And a green note	5
 Anything but default colors	6
 A note without a shadow	6
 A note with a shadow	6
 Test of option.	6
 A note with no line connecting it to the placement in the original text.	6
 A todonote placed in the text	6
 Fill those circles ...	6
 A note with a large font size.	7
 Note with very small font size.	7
 Short note	7
 Short note with prepend	7
 Short note with noprepnd	7
 Testing.	7
 Testing author option.	7
 Testing author option.	7
 Testing the option inlinewidth.	7
 Testing the option inlinepar.	8
Figure: Testing a long text string	8
Figure: Testing a long text string	8
Figure: Add an image ...	8
Figure: Testing	9
Figure: Testing figcolor	9
 Test of newly defined command.	14
 Test of newly defined command, requesting a green color.	14
 1: A numbered todonote.	14
 2: Another numbered todonote.	14

	Comment [HSM1]: Testing first time.	15
	Comment [HSM2]: Testing second time.	15
	Some lines with a decreased line spacing. This is accomplished using the setspace package that is included in standard latex distributions. . . .	16
	Some lines with a decreased line spacing. This is accomplished without using any special packages.	16
	2do	17
	Translation	17
	Translation	17
	1.8.9. A numbered todo.	18
	1. Small notes with links back to the list of todos.	18
	1. Smart notes with links back to the list of todos.	18
	fix text	19
	Testing	19
	Testing	19
	Testing	19
	Testing	19

`\todototoc` The `\todototoc` command adds an entry to the table of contents for list of todos. The command should be placed right before the `\listoftodos` command.

1.2 Package options

`disable` If the option `disable` is passed to the package, the macros usually defined by the package (`\todo`, `\listoftodos` and `\missingfigure`) are defined as macros with no effect, and thus all inserted notes are removed.

`obeyDraft`, `obeyFinal` When the option `obeyDraft` is given, the package checks if the one of the options `draft`, `draftcls` or `draftclsnofoot` is given (this option is usually given to the documentclass). If the `draft` option is given, the functionality of the package is enabled and otherwise the effect of the package is disabled. The option `obeyFinal` does something similar, except that the `todonotes` package is only disabled if the `final` option given.

`danish`, `german`, `ngerman`,
`english`, `french`, `swedish`
`spanish`, `catalan`, `italian`
`portuguese`, `dutch`,
`croatian`
`colorinlistoftodos` Use translations of the text strings "List of todos" and "Missing figure". The default is to use none of these options, which results in english text strings. Currently the following languages are supported: catalan, croatian, danish, dutch, english, french, german, ngerman, italian, portuguese, spanish and swedish.

Adds a small colored square in front of all items in the Todo list. The color of the square is the same as the fill color of the inserted todonote. This can be useful if there are different types of todos (insert reference, explain in detail, ...) where the color of the inserted todonote marks the type of todo.

`color` These options sets the default colors for the todo command. There are four colors that can be specified. The border color (default `bordercolor=black`) around the inserted text, the color behind the inserted text (default `backgroundcolor=orange`), the color of the inserted text (default `textcolor=black`) and the color of the line connecting the inserted textbox with the current location in the text (default `bordercolor`)

linecolor=orange). Setting the color option to val passes this value on to the background and line color options. The specified colors must be valid according to the xcolor package.

tickmarkheight	tickmarkheight=length set the height of the tickmark at the location where the todo is inserted, the default value is 0pt.
textwidth	textwidth=length sets the width of a todo item in the margin to length. The width of inline todonotes will always be the same as the current line width.
textsize	textsize=value sets the default text size of the inserted todonotes to the given value. Value is the "name" of the used font size, eg. if the desired fontsize is \tiny use textsize=tiny. The default value is textsize=normalsize.
prependcaption	The prependcaption option triggers a special behaviour of the caption=val option for the todo command, where the given value val is inserted in the inserted todonote.
loadshadowlibrary	If given, the shadows tikz library will be loaded. This is required for adding shadows to the inserted todonotes.
shadow	If the shadow option is given, the inserted todonotes will by default be displayed with a gray shadow. For this to work, the package option loadshadowlibrary must be provided. I expect that the option will trigger problems with tikz versions prior to 2.0.
dvistyle	When a document with todonotes is compiled with plain latex (to a dvi-file), there is an issue with the visual appearance ¹ . The option dvistyle changes the appearance of the inserted todonotes to avoid this problem.
figwidth	The figwidth=length option sets the default width of the figure inserted by the \missingfigure command. The default value is \linewidth.

1.3 Options for the todo command

There are several options that can be given to the \todo command. All the options are described here and often I have included examples of the change in visual appearance. Default values for these options can be set using the setupodonotes command.

```
\setupodonotes{fancyline, color=blue!30}
```

disable	The disable option can be given directly to the todo command. If given the command has no effect.
color	These options set the color that is used in the current todo command. The color classes is the same as used in the color package options, see section 1.2.
backgroundcolor	Default values can be set by the color options when the todonotes package is loaded. The todo notes inserted in this paragraph is created with the command
textcolor	\todo[color=green!40]{And a green note}. The color of the inserted note
linecolor	could be used to mark different types of tasks (insert references, explain something in detail, ...), this could be streamlined by defining new commands like below.
bordercolor	

```
\newcommand{\insertref}[1]{\todo[color=green!40]{#1}}
```

¹The problem is placement of text inside the colored boxes.

And a green note

```
\newcommand{\explainindetail}[1]{\todo[color=red!40]{#1}}
```

Anything but default colors

An example that uses all of the color options is given below.

```
\todo[linecolor=green!70!white, backgroundcolor=blue!20!white,
bordercolor=red, textcolor=yellow]{Anything but default colors}.
```

shadow / noshadow

If the package is loaded with the `shadow` option, a shadow is added to all notes, otherwise no shadows are shown. This global setting can be overwritten by the `shadow` and `noshadow` options to the `todo` command. To insert a note with and without shadows, use the `shadow` and `noshadow` options as shown here:

A note without a shadow

A note with a shadow

```
\todo[shadow]{A note with a shadow}
\todo[noshadow]{A note without a shadow}
```

tickmarkheight

Test of option.

`tickmarkheight=length` set the height of the tickmark at the location where the `todo` is inserted. An example of the option in use is shown here.

```
\todo[tickmarkheight=0.1cm]{Test of option.}
```

line / noline

A note with no line connecting it to the placement in the original text.

If you want to get rid of the line connecting the inserted note with the place in the text where the note occurs in the latex code, the option `noline` can be used.

```
\todo[noline]{A note with no line ...}
```

inline / noinline

It is possible to place a `todonote` inside the text instead of placing it in the margin, this could be desirable if the text in the note has a considerable length.

```
\todo[inline]{A todonote placed in the text}
```

A todonote placed in the text

Another usage for the `inline` option is when you want to add a `todonote` to a figure caption.

```
\begin{wrapfigure}{r}[20mm]{40mm}
\begin{tikzpicture}
\draw[red] (0, 0) circle(0.45);
\draw[green] (1, 0) circle(0.45);
\draw[blue] (2, 0) circle(0.45);
\end{tikzpicture}
\caption{A text explaining the image.}
\todo[inline]{Fill those circles \ldots}
\end{wrapfigure}
```

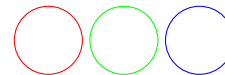


Figure 1: A text explaining the image.

Fill those circles ...

`size` `size=val` changes the size of the text inside the `todonote`. The commands used to create the notes below are

A note with a large font size.

list / nolist

`\todo[size=\Large]{A note with a large font size.}` and `\todo[inline, size=\tiny]{Note with very small font size.}`.

Note with very small font size.

When the option `nolist` is given, the `todo` item will not appear in the list of todos.

caption

The `caption` option enables the user to specify a short description of the `todonote` that are inserted in the list of todos instead of the full `todonote` text.

A very long and tedious note that cannot be on one line in the list of todos.

`\todo[caption={Short note}]{A very long and tedious note that cannot be on one line in the list of todos.}`.

The effect of this option is altered with the package option `prependcaption` or the `prepend / noprepend` option for the `todo` command.

prepend / noprepend

The options `prepend` and `noprepend` can be used for setting whether a given caption should be prepended to the `todonote` or not. Globally this can be set using the `prependcaption` option for the package. Below is the effect of the option shown using the code:

Short note with prepend:
A very long and tedious note that cannot be on one line in the list of todos.

`\todo[prepend, caption={Short note with prepend}]{A very long and tedious note that cannot be on one line in the list of todos.}`
`\todo[noprepend, caption={Short note with noprepend}]{A very long and tedious note that cannot be on one line in the list of todos.}`

A very long and tedious note that cannot be on one line in the list of todos.

The `fancyline` option inserts a curved arrow, pointing from the inserted note to the insertion point. The option is used like this:

fancyline

Testing.

`\todo[fancyline]{Testing.}`

author

The `author` option takes a parameter, the name of the author. The given name is inserted in the `todonote`.

Xavier

Testing author option.

Xavier: Testing author option.

`\todo[author=Xavier]{Testing author option.}`
`\todo[author=Xavier, inline]{Testing author option.}`

inlinewidth

The `inlinewidth` option makes it possible to alter the width of inline todos.

Testing the option inlinewidth.

`\todo[inline, inlinewidth=5cm]{Testing the option inlinewidth.}`

inlinepar / noinlinepar

The options `inlinepar` and `noinlinepar` (default) determine whether the inserted inline `todonote` is surrounded with `par` macros that enforce new lines

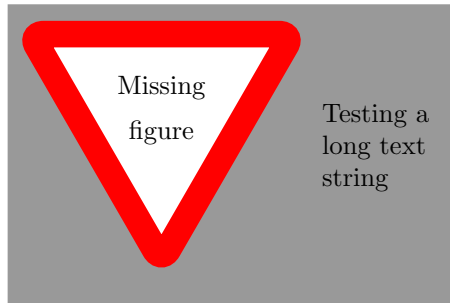
prior to and after inserted inline todonotes. Testing the option inlinepar. This is a line after the inserted todonote.

```
\todo[inline, inlinewidth=5cm, noinlinepar]{Testing the option inlinepar.}
```

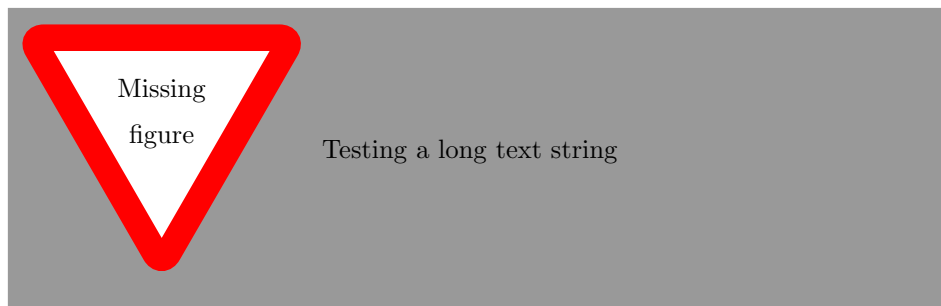
1.4 Options for the missingfigure command

figwidth The `figwidth=length` option sets the width of the figure inserted by the `\missingfigure` command. Length values below *6cm* might trigger some problems with the visual appearance. Try to compare the default of the missing figure command, when the option is given or not.

```
\missingfigure[figwidth=6cm]{Testing a long text string}
```

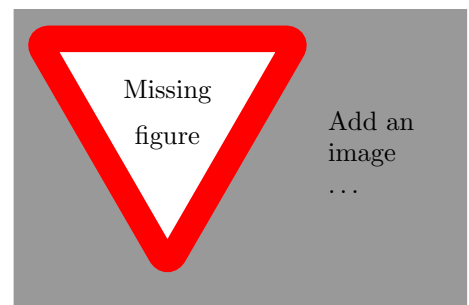


```
\missingfigure{Testing a long text string}
```



Another usage of the option is when `\missingfigure` is used in the `wrapfigure` environment.

```
\begin{wrapfigure}[1]{r}[2cm]{6cm}  
\missingfigure[figwidth=6cm]{Add an image \ldots}  
\end{wrapfigure}
```



figheight The `figheight=length` option changes the height of the inserted missing figure. The default height is 4cm and using values lower than this might cause the warning sign to pop out of the gray area.

```
\missingfigure[figheight=6cm]{Testing a long text string}
```



figcolor The `figcolor=color` options sets the background color of inserted missing figures. The default color is `black!40`.

```
\missingfigure[figcolor=white]{Testing figcolor}
```



1.5 Options for the `listoftodos` command

The `\listoftodos` command takes one optional argument, that defines the name of the inserted list of todos.

```
\listoftodos[I can be called anything]
```

1.6 Known issues

1.6.1 Package loading order

The todonotes package requires the following packages.

- ifthen
- xkeyval
- xcolor
- tikz
- calc
- graphicx (is loaded via the tikz package)

When todonotes are loaded in the preamble, the package checks if these packages all are loaded. If that is not the case it loads the missing packages with no options given. If you want to give some specific options to some of these packages, you have to load them *before* the todonotes package, otherwise you will get an "Option clash" error when latex works on the document.

If both the menukeys and the xcolor (with the option `table`) package should be loaded, the following order must be used.

```
\usepackage[table]{xcolor}
\usepackage{todonotes}
\usepackage{menukeys}
```

The following packages must be loaded before the todonotes package:

- polyglossia

1.6.2 Wrapping of long lines in list of todos

When a document is compiled with latex (and not pdflatex) long items in the list of todos are not wrapped into several lines, and do instead continue to the right out of the page.

1.6.3 Conflicts with the amsart documentclass

The `amsart` document class redefines some internal commands that is used by the todonotes package, this will cause an malfunctioning `\listoftodos` command. The following code to circumvent the problem was given by Dan Luecking on `comp.text.tex`

```
\makeatletter
\providecommand\@dotsep{5}
\makeatother
\listoftodos\relax
```

NOT TESTED NOT TESTED NOT TESTED
Dominique suggests the following workaround.

```
\makeatletter
\providecommand\@dotsep{5}
\def\listtodoname{List of Todos}
\def\listoftodos{\@starttoc{tdo}\listtodoname}
\makeatother
```

1.6.4 Unknown option "remember picture"

If latex throws the error

```
Package tikz Error: I do not know what to do with the option ‘remember picture’.
```

It probably means that your latex installation is outdated, as only newer versions of latex driver for tikz supports the `remember picture` option. For additional info consult "Section 9.2.2 Producing PDF Output" in the tikz manual. <http://www.ctan.org/tex-archive/graphics/pgf/base/doc/generic/pgf/pgfmanual.pdf>

1.6.5 Todonotes wrongly placed in the margin

When using some document classes or packages, the todonotes inserted in the page margin can be placed quite oddly. This is often caused by a wrong value of the `\marginparwidth` length. Try using the code below in your preamble to see if this cures the problem.

```
\setlength{\marginparwidth}{2cm}
```

If the todonotes are inserted in the wrong margin, the solution is the `\reversemarginpar` command. When this command is issued the following marginpars (which todonotes relies on) is inserted in the other margin.

1.6.6 Reduce number of warnings

If the width of the inserted todonotes is forced to be larger than the available space in the margin, a lot of warnings are issued. This can be reduced with the following code.

```
\usepackage[textwidth=3.7cm]{todonotes}
\setlength{\marginparwidth}{3.7cm}
```

1.6.7 Todonotes in footnotes

Placing todonotes in footnotes throws a lot of errors. Richard Stanton comes with the following work around.

```
\renewcommand{\marginpar}{\marginnote}
```

1.6.8 Trouble with classicthesis.sty

[Problem description on tex.stackexchange.com](http://tex.stackexchange.com).

Solution by Stefan Kottwitz. The problem is caused by the redefinition of `\marginpar` in `classicthesis.sty`. `\marginpar` is used by todonotes. It can be fixed by restoring the original meaning, if you don't need the classicthesis marginpar style. Just add this to your document preamble: `\let\marginpar\oldmarginpar`

1.6.9 Todos in math and floating environments

It is not possible to insert todonotes into math environments or floating environments like `figure` or `table`. By replacing the `marginpar` command with a `marginnote` this can be enabled. For todos inside math environments the todo must be placed in an `mbox` or `\textrm` command.

```
\documentclass{article}
\usepackage{todonotes}
\usepackage{marginnote}
\let\marginpar\marginnote

\begin{document}
\begin{equation}
a^2=b^2+c^2 \ \textrm{\todo{Test}}
\end{equation}
\begin{figure}
\centering
\caption{Caption\todo{tests}.}
\label{fig}
\end{figure}
```

1.6.10 Conflict between marginpar and the standalone class

The `standalone` document class is not compatible with the `\marginpar` command, which is used in the todonotes package to insert contents in the side margins. A workaround is to replace the `\marginpar` with `\marginnote`. This is demonstrated here:

```
\documentclass[crop=false]{standalone}
\usepackage{todonotes}
\usepackage{marginnote}
```

```
\let\marginpar\marginnote
\begin{document}
Lorem ipsum
\todo{Bug}
\end{document}
```

1.7 Things to improve

This is a list of things I consider to improve sometime in the future. It have not been done yet as I lack the time or skills to implement them. Patches with implementations of these tasks will be appreciated and might be included in the package if it will improve the package quality.

1.7.1 Owner information

Option for the todo command.

```
\todo[owner={Fabrice}]{Stuff}
```

Add info on who "owns" the current todo. Idea: Fabrice Niessen

1.7.2 Due date

Option for the todo command.

```
\todo[due=2008-12-07]{Stuff}
```

Add info on when the current todo is due. Might be enhanced by a time line of the todos that have a due date assigned. Idea: Fabrice Niessen

1.7.3 Mark accomplished todos

```
\todo[done]{Stuff}
```

Idea: Fabrice Niessen

1.8 Usage methods

In this section I have collected some different methods to use the `todonotes` package.

1.8.1 Define new commands with fixed options

Often there is a need for marking different classes of things to do (add reference, rewrite, ...). One way to do this, is to define some new commands as shown below (idea from Florent B.).

```
\newcommand{\addref}{\todo[color=red!40]{Add reference.}}
\newcommand{\rewrite}[1]{\todo[color=green!40]{#1}}
```

To distinguish between the different types of todos, the `todonotes` package can be loaded with the `colorinlistoftodos` option, which adds small colored squares to the list of todos.

```
\usepackage[colorinlistoftodos]{todonotes}
```

1.8.2 Define new commands with arbitrary default options

If you do not like the default values of the standard `\todo` command, it is possible to define a new command with the similar functionality of `\todo` with custom default values.

```
\newcommand{\todoredefined}[2] []
{\todo[color=red, #1]{#2}}
```

Test of newly defined command.

The new command can now be used like shown below

```
\todoredefined{Test of newly defined command.}
\todoredefined[color=green]{Test of newly defined command, requesting a green color.}
```

Test of newly defined command, requesting a green color.

This can be done with all the accepted options for the `\todo` command.

1.8.3 Enumerate todonotes

If the inserted todonotes should be enumerated, it is possible to define a new command with the desired behaviour.

```
\newcounter{todocounter}
\newcommand{\todonum}[2] []
{\stepcounter{todocounter}\todo[#1]{\thetodocounter: #2}}
```

1: A numbered todonote.

The idea is to define a new counter `todocounter`, and insert the value of the counter in each todonote. The new command can be used like

2: Another numbered todonote.

```
\todonum{A numbered todonote.}
\todonum{Another numbered todonote.}
```

1.8.4 Comments "a la Word"

Fabrice Niessen sent me the following use case. The idea is to define a new command `\mycomment` which adds a counter and optionally the initials of the author to the inserted todonote.

```
\newcounter{mycomment}
\newcommand{\mycomment}[2] []{%
  % initials of the author (optional) + note in the margin
  \refstepcounter{mycomment}%
  {%
    \setstretch{0.7}% spacing
    \todo[color={red!100!green!33},size=\small]{%
      \textbf{Comment [\uppercase{#1}\themycomment]:~#2}%
    }
  }}

```

Comment [HSM1]: Testing first time.

The command `\mycomment[HSM]{Testing first time.}` is displayed like shown in the left margin, and another call of the command is added below `\mycomment[HSM]{Testing second time.}`.

Comment [HSM2]: Testing second time.

1.8.5 Combination with the `fixme` package

Thomas Arildsen has mailed me this use case. Check the documentation for the `fixme` package, as the code below relies directly on it (the `\FDUser` command is augmented when `\begin{document}` is reached).

```
\usepackage[user,nomargin]{fixme}
\usepackage{todonotes}
\newcommand{\FXUser}[2]{\todo[inline,size=\small]{\bfseries #1:} #2}}

```

1.8.6 Altering the line spacing of todonotes

The `setspace` package lets you alter the line spacing of smaller sections of your document. The primary construct is the `spacing` environment, which is demonstrated below.

```
\begin{spacing}{0.5}
Some lines with a decreased line spacing. This is accomplished
using the setspace package that is included in standard latex
distributions.
\end{spacing}

```

Some lines with a decreased line spacing. This is accomplished using the `setspace` package that is included in standard latex distributions.

Using the `spacing` environment we can define a new todonote command using the code below:

```
\newcommand{\smalltodo}[2] []

```

```
{\todo[caption={#2}, #1]
{\begin{spacing}{0.5}#2\end{spacing}}}
```

Some lines with a decreased line spacing. This is accomplished using the setspace package that is included in standard latex distributions.

Todonotes with decreased line spacing can now be inserted as follows

```
\smalltodo[size=\footnotesize]{
Some lines with a decreased line spacing. This is accomplished
using the setspace package that is included in standard latex
distributions.}
```

A different approach is given by Vitaly.

```
\newcommand{\tinytodo}[2] []
{\todo[caption={#2}, size=\small, #1]{\renewcommand{\baselinestretch}{0.5}\selectfont#2\par}}
```

Some lines with a decreased line spacing. This is accomplished without using any special packages.

It looks like seen here.

```
\tinytodo{
Some lines with a decreased line spacing. This is accomplished
without using any special packages.}
```

1.8.7 Marking new / old sections

Sometimes a whole section has to be marked by some means. You might want to try the following.

```
\todo[inline, caption={Some text}]{
\begin{minipage}{\linewidth}
Some text that might differ from the text given to the caption
option.
\end{minipage}
}
```

It is important to add the `caption={text}` option, otherwise latex will try to embed a minipage in the table of contents which triggers an error. Inside the minipage environment almost anything could be placed, except for other todo commands.

To streamline use the following command was suggested by Stefan Pinnow.

```
\newcommand\todoIn[2] []{\todo[inline, caption={2do}, #1]{
\begin{minipage}{\textwidth-4pt}#2\end{minipage}}}
```

This example renders like

```
\todoIn{
Some text.}
```



```

\begin{align}
\sin(\theta)^2 + \cos(\theta)^2 = 1
\end{align}
A formula and a list
\begin{itemize}
\item An item
\end{itemize}
}

```

Some text.

$$\sin(\theta)^2 + \cos(\theta)^2 = 1 \tag{1}$$

A formula and a list

- An item

1.8.8 Link to list of todos

Using the `hyperref` package it is possible to add a link from the inserted todonotes to the list of todos. The example were supplied by Andreas Plank.

```

% Define a counter for the inserted todonotes.
\newcounter{todoListItems}
\newcommand{\todoTrans}[2][ ]{
  % Increment counter
  \addtocounter{todoListItems}{1}
  \todo[
    caption={\protect\hypertarget{todo\thetodoListItems}{Translation},
    #1]
  {
    #2 \hfill
    \hyperlink{todo\thetodoListItems}{\uparrow}
  }
}

```

The idea behind the code is to embed a `hypertarget` in each entry in the list of todos. In the todonotes a link to the entry in the list of todos is inserted as an arrow that points upwards. Using the `\todoTrans` command like below, the following two notes have been inserted.

```

\todoTrans{papiersflyver}
\todoTrans[inline]{damplokomotiv}

```

papiersflyver



damplokomotiv



1.8.9 Numbered todonotes

The inserted todonotes can be argumented with the current subsubsection number. The code is shown below.

```
\newcommand{\ntodo}[2] [] {\todo[#1]{\thesubsubsection{}. #2}}
```

By changing `\thesubsubsection` to `\thesection`, the current section number can be inserted instead of the subsubsection number. The result looks like. Which were generated by the code

```
\ntodo{A subsection numbered todo.}.
```

1.8.10 Combining several modifications

Manduca have combined several of the modifications above into a highly specialized todo command. She uses the code:

```
\newcounter{todoListItems}
\newcommand{\sstodo}[2] []
{\addtocounter{todoListItems}{1}
\todo[caption={\protect\hypertarget{todo\thetodoListItems}{}\thesection. #2}, #1]
{\begin{spacing}{1} \hfill \hyperlink{todo\thetodoListItems}{#2} \end{spacing} }}
```

Using this approach it is possible to customize the behavior of the inserted notes to a very high degree.

1.8.11 Alter the appearance of the list of todos

Marco Daniel gives the following example of how to add section numbers to the elements in the list of todos. The code is slightly modified from <http://tex.stackexchange.com/questions/18838/replacing-page-number-with-other-counter-in-listoftodos>. An example of the modified list of todos is shown below, the complete code example is given in the example directory.

```
This is a note . . . . . see 1.1 at p. 2
This is another note . . . . . see 1.2 at p. 4
```

1.8.12 Tikz externalization issues

Using the tikz externalization framework together with todonotes can lead to some problems. One solution is to disable the externalization just before the `todo` command is issued and then reactivate externalization afterwards. The `ruggedtodo` handles this deactivation and reactivation.

```
\usetikzlibrary{external}
\tikzexternalize
\newcommand{\ruggedtodo}[2] [] {\tikzexternaldisable\todo[#1]{#2}\tikzexternalenable}
```

1.8.9. A numbered todo.

Small notes with links back to the list of todos.

Smart notes with links back to the list of todos.

1.8.13 Highlight text to fix

Tobias Winchen provides the following example on how to highlight text related to the inserted todonote. Example `wrong text` continues here. Notice that the code relies on the soul package.

fix text

```
\makeatletter
\if@todonotes@disabled
\newcommand{\hlfix}[2]{#1}
\else
\newcommand{\hlfix}[2]{\texthl{#1}\todo{#2}}
\fi
\makeatother
Example \hlfix{wrong text}{fix text}~continues here.
```

1.8.14 Adding color coded levels of todos

Sometimes it can be beneficial to be able to distinguish between different types of todos. For this I would suggest the following approach:

```
\newcommand{\todoerror}[2] []{\todo[color=red!70, #1]{#2}}
\newcommand{\todowarn}[2] []{\todo[color=orange, #1]{#2}}
\newcommand{\todoremark}[2] []{\todo[color=yellow!80!black, #1]{#2}}
\newcommand{\todohint}[2] []{\todo[color=green!20, #1]{#2}}
\newcommand{\tododone}[2] []{\todo[color=blue!50, #1]{#2}}

\todoerror[inline]{Testing}
\todoremark{Testing}
\todowarn[inline]{Testing}
\todohint{Testing}
\tododone[disable]{Testing}
```

Testing

Testing

Testing

Testing

2 Implementation

Identifies the package and loads the packages dependences.

```
1 \ProvidesPackage{todonotes}[2021/06/04]
2 \RequirePackage{ifthen}
3 \RequirePackage{xkeyval}
4 \RequirePackage{xcolor}
5 \RequirePackage{tikz}
6 \usetikzlibrary{positioning}
7 \RequirePackage{calc}
```

Implement a function for setting up the todonotes package.

```
8 \newcommand\setuptodonotes[1]{\presetkeys{todonotes}{#1}{}}
```

Some default values are set

```
9 \newcommand{\@todonotes@text}{}%
10 \newcommand{\@todonotes@backgroundcolor}{orange}
11 \newcommand{\@todonotes@textcolor}{black}
12 \newcommand{\@todonotes@linecolor}{orange}
13 \newcommand{\@todonotes@bordercolor}{black}
14 \newcommand{\@todonotes@tickmarkheight}{unused value}
15 \newcommand{\@todonotes@textwidth}{\marginparwidth}
16 \newcommand{\@todonotes@textsize}{\normalsize}
17 \newcommand{\@todonotes@figwidth}{\linewidth}
18 \newcommand{\@todonotes@figheight}{4cm}
19 \newcommand{\@todonotes@figcolor}{black!40}

20 \AtBeginDocument{
21 \ifx\undefined\phantomsection
22 \newcommand{\phantomsection}{}
23 \fi
24 \ifdim \marginparwidth < 2cm
25 \PackageWarning{todonotes}{The length marginparwidth is
26 less than 2cm and will most likely cause issues with the
27 appearance of inserted todonotes.
28 The issue can be solved by adding a line like
29 \setlength{\marginparwidth}{2cm}
30 prior to loading the todonotes package.} \else\fi%
31 }
```

2.1 Declaration of options for the package

In this part the various options for the package are defined.

Define the default text strings and set localization options for the danish and german languages.

```
32 \newcommand{\@todonotes@todolistname}{Todo list}
33 \newcommand{\@todonotes@MissingFigureText}{Figure}
34 \newcommand{\@todonotes@MissingFigureUp}{Missing}
35 \newcommand{\@todonotes@MissingFigureDown}{figure}
36 \newcommand{\@todonotes@SetTodoListName}[1]
```

```

37   {\renewcommand{\@todonotes@todolistname}{#1}}
38 \newcommand{\@todonotes@SetMissingFigureText}[1]
39   {\renewcommand{\@todonotes@MissingFigureText}{#1}}
40 \newcommand{\@todonotes@SetMissingFigureUp}[1]
41   {\renewcommand{\@todonotes@MissingFigureUp}{#1}}
42 \newcommand{\@todonotes@SetMissingFigureDown}[1]
43   {\renewcommand{\@todonotes@MissingFigureDown}{#1}}
44 \newif{\if@todonotes@reverseMissingFigureTriangle}
45 \DeclareOptionX{catalan}{%
46   \@todonotes@SetTodoListName{Llista de feines pendants}%
47   \@todonotes@SetMissingFigureText{Figura}%
48   \@todonotes@SetMissingFigureUp{Figura}%
49   \@todonotes@SetMissingFigureDown{pendent}%
50 }
51 \DeclareOptionX{croatian}{%
52   \@todonotes@SetTodoListName{Popis obveza}%
53   \@todonotes@SetMissingFigureText{Slika}%
54   \@todonotes@SetMissingFigureUp{Nedostaje}%
55   \@todonotes@SetMissingFigureDown{slika}%
56 }
57 \DeclareOptionX{danish}{%
58   \@todonotes@SetTodoListName{G\o{}rem\aa{}lsliste}%
59   \@todonotes@SetMissingFigureText{Figur}%
60   \@todonotes@SetMissingFigureUp{Manglende}%
61   \@todonotes@SetMissingFigureDown{figur}%
62 }
63 \DeclareOptionX{dutch}{%
64   \@todonotes@SetTodoListName{Lijst van onafgewerkte taken}%
65   \@todonotes@SetMissingFigureText{Figuur}%
66   \@todonotes@SetMissingFigureUp{Ontbrekende}%
67   \@todonotes@SetMissingFigureDown{figuur}%
68 }
69 \DeclareOptionX{english}{%
70   \@todonotes@SetTodoListName{Todo list}%
71   \@todonotes@SetMissingFigureText{Figure}%
72   \@todonotes@SetMissingFigureUp{Missing}%
73   \@todonotes@SetMissingFigureDown{figure}%
74 }
75 \DeclareOptionX{french}{%
76   \@todonotes@SetTodoListName{Liste des points \`a traiter}%
77   \@todonotes@SetMissingFigureText{Figure}%
78   \@todonotes@SetMissingFigureUp{Figure}%
79   \@todonotes@SetMissingFigureDown{manquante}%
80   \@todonotes@reverseMissingFigureTrianglefalse
81 }
82 \DeclareOptionX{german}{%
83   \@todonotes@SetTodoListName{Liste der noch zu erledigenden Punkte}%
84   \@todonotes@SetMissingFigureText{Abbildung}%
85   \@todonotes@SetMissingFigureUp{Fehlende}%
86   \@todonotes@SetMissingFigureDown{Abbildung}%

```

```

87 }
88 \DeclareOptionX{italian}{
89   \@todonotes@SetTodoListName{Elenco delle cose da fare}%
90   \@todonotes@SetMissingFigureText{Figura}%
91   \@todonotes@SetMissingFigureUp{Figura}%
92   \@todonotes@SetMissingFigureDown{mancante}%
93 }
94 \DeclareOptionX{ngerman}{%
95   \@todonotes@SetTodoListName{Liste der noch zu erledigenden Punkte}%
96   \@todonotes@SetMissingFigureText{Abbildung}%
97   \@todonotes@SetMissingFigureUp{Fehlende}%
98   \@todonotes@SetMissingFigureDown{Abbildung}%
99 }
100 \DeclareOptionX{portuguese}{
101   \@todonotes@SetTodoListName{Lista de tarefas pendentes}%
102   \@todonotes@SetMissingFigureText{Figura}%
103   \@todonotes@SetMissingFigureUp{Figura}%
104   \@todonotes@SetMissingFigureDown{pendente}%
105 }
106 \DeclareOptionX{spanish}{
107   \@todonotes@SetTodoListName{Lista de tareas pendientes}%
108   \@todonotes@SetMissingFigureText{Figura}%
109   \@todonotes@SetMissingFigureUp{Figura}%
110   \@todonotes@SetMissingFigureDown{pendiente}%
111 }
112 \DeclareOptionX{swedish}{%
113   \@todonotes@SetTodoListName{Att g\{"o}ra-lista}%
114   \@todonotes@SetMissingFigureText{Figur}%
115   \@todonotes@SetMissingFigureUp{Figur}%
116   \@todonotes@SetMissingFigureDown{saknas}%
117 }

```

Create a counter, for storing the number of inserted todos.

```
118 \newcounter{@todonotes@numberoftodonotes}
```

Toggle whether the package should obey the global draft option.

```

119 \newif{\if@todonotes@obeyDraft}
120 \DeclareOptionX{obeyDraft}{\@todonotes@obeyDrafttrue}
121 \newif{\if@todonotes@isDraft}
122 \DeclareOptionX{draft}{\@todonotes@isDrafttrue}
123 \DeclareOptionX{draftcls}{\@todonotes@isDrafttrue}
124 \DeclareOptionX{draftclsnofoot}{\@todonotes@isDrafttrue}
125 \newif{\if@todonotes@obeyFinal}
126 \DeclareOptionX{obeyFinal}{\@todonotes@obeyFinaltrue}
127 \newif{\if@todonotes@isFinal}
128 \DeclareOptionX{final}{\@todonotes@isFinaltrue}

```

Make it possible to disable the functionality of the package. If this option is given, the commands `\todo{}` and `\listoftodos` are defined as commands with no effect. (But you can still compile you document with these commands).

```
129 \newif{\if@todonotes@disabled}
```

```

130 \DeclareOptionX{disable}{\@todonotes@disabledtrue}
    Show small boxes in the list of todos with the color of the inserted todonotes.
131 \newif{\if@todonotes@colorinlistoftodos}
132 \DeclareOptionX{colorinlistoftodos}{\@todonotes@colorinlistoftodostrue}
    The default style behaves bad when compiled with latex (some text placement
    problems). The dvistyle option, changes the visual behavior to avoid this text
    placement problem.
133 \newif{\if@todonotes@dviStyle}
134 \DeclareOptionX{dvistyle}{\@todonotes@dviStyletrue}
    Create a color option.
135 \define@key{todonotes.sty}%
136     {color}{
137         \renewcommand{\@todonotes@backgroundcolor}{#1}
138         \renewcommand{\@todonotes@linecolor}{#1}}
    Make the background color of the notes as an option.
139 \define@key{todonotes.sty}%
140     {backgroundcolor}{\renewcommand{\@todonotes@backgroundcolor}{#1}}
    Make the text color of the notes as an option.
141 \define@key{todonotes.sty}%
142     {textcolor}{\renewcommand{\@todonotes@textcolor}{#1}}
    Make the line color of the notes as an option.
143 \define@key{todonotes.sty}%
144     {linecolor}{\renewcommand{\@todonotes@linecolor}{#1}}
    Make the color of the notes box color as an option.
145 \define@key{todonotes.sty}%
146     {bordercolor}{\renewcommand{\@todonotes@bordercolor}{#1}}
    Make the height of the line start marking as an option.
147 \newcommand{\@todonotes@defaulttickmarkheight}{0cm}
148 \define@key{todonotes.sty}{tickmarkheight}{%
149     \renewcommand{\@todonotes@defaulttickmarkheight}{#1}}%
    Set whether short captions given as arguments to the todo command should be
    included in the inserted todonote.
150 \newif{\if@todonotes@prependcaptionglobal}
151 \@todonotes@prependcaptionglobalfalse
152 \DeclareOptionX{prependcaption}{\@todonotes@prependcaptionglobaltrue}
    Make the text width as an option.
153 \define@key{todonotes.sty}%
154     {textwidth}{\renewcommand{\@todonotes@textwidth}{#1}}
    Make the text size as an option, accept both size and textsize.
155 \define@key{todonotes.sty}%
156     {textsize}{\renewcommand{\@todonotes@textsize}{#1}}
157 \define@key{todonotes.sty}%
158     {size}{\renewcommand{\@todonotes@textsize}{#1}}

```

Add option for shadows behind the inserted notes

```
159 \newif\if@todonotes@shadowlibraryloaded
160 \@todonotes@shadowlibraryloadedfalse
161 \DeclareOptionX{loadshadowlibrary}{%
162   \usetikzlibrary{shadows}%
163   \@todonotes@shadowlibraryloadedtrue}
164 \newcommand{\@todonotes@shadowenabledbydefault}{noshadow}
165 \DeclareOptionX{shadow}{%
166   \renewcommand{\@todonotes@shadowenabledbydefault}{shadow}}
```

Add option for the default width of the figure inserted with `\missingfigure`.

```
167 \define@key{todonotes.sty}%
168   {figwidth}{\renewcommand{\@todonotes@figwidth}{#1}}
169 \define@key{todonotes.sty}%
170   {figheight}{\renewcommand{\@todonotes@figheight}{#1}}
171 \define@key{todonotes.sty}%
172   {figcolor}{\renewcommand{\@todonotes@figcolor}{#1}}
```

Make the text width as an option.

```
173 % Finally process the given options.
174 %   \begin{macrocode}
175 \ProcessOptionsX*
```

If the `obeyDraft` is given, check whether one of the `draft`, `draftcls` or `draftclsnofoot` options are given and enable or disable the functionality of this package. If the `obeyFinal` option is given together with the `final` option the `todonotes` are disabled. The `disable` option will overrule the effect of `obeyDraft`.

```
176 \if@todonotes@disabled
177 \else
178 \if@todonotes@obeyDraft
179 \@todonotes@disabledtrue
180 \if@todonotes@isDraft
181 \@todonotes@disabledfalse
182 \fi
183 \fi
184 \if@todonotes@obeyFinal
185 \@todonotes@disabledfalse
186 \if@todonotes@isFinal
187 \@todonotes@disabledtrue
188 \fi
189 \fi
190 \fi
```

2.2 Options for the `todo` command

In this part the various options for commands in the package are defined. Set an arbitrarily fill color

```
191 \gdef\@todonotes@currentlinecolor{\@todonotes@linecolor}%
192 \gdef\@todonotes@currentbackgroundcolor{\@todonotes@backgroundcolor}%
193 \gdef\@todonotes@currenttextcolor{\@todonotes@textcolor}%
```



```

194 \gdef\@todonotes@currentbordercolor{\@todonotes@bordercolor}%
195 \define@key{todonotes}{color}{%
196   \gdef\@todonotes@currentlinecolor{#1}%
197   \gdef\@todonotes@currentbackgroundcolor{#1}}%
198 \define@key{todonotes}{linecolor}{%
199   \gdef\@todonotes@currentlinecolor{#1}}%
200 \define@key{todonotes}{backgroundcolor}{%
201   \gdef\@todonotes@currentbackgroundcolor{#1}}%
202 \define@key{todonotes}{textcolor}{%
203   \gdef\@todonotes@currenttextcolor{#1}}%
204 \define@key{todonotes}{bordercolor}{%
205   \gdef\@todonotes@currentbordercolor{#1}}%

Toggle whether there is a shadow behind the inserted notes.
206 \newif\if@todonotes@useshadow%
207 \define@key{todonotes}{shadow}[]{\@todonotes@useshadowtrue}%
208 \define@key{todonotes}{noshadow}[]{\@todonotes@useshadowfalse}%

Define height of the inserted tickmark.
209 \define@key{todonotes}{tickmarkheight}{%
210   \renewcommand{\@todonotes@tickmarkheight}{#1}}%

Set a relative font size
211 \newcommand{\@todonotes@sizecommand}{%
212 \define@key{todonotes}{size}{\renewcommand{\@todonotes@sizecommand}{#1}}%
213 }%

Should the todo item be disabled?
214 \newif\if@todonotes@localisable%
215 \define@key{todonotes}{disable}[]{\@todonotes@localisabletrue}%
216 \define@key{todonotes}{nodisable}[]{\@todonotes@localisablefalse}%

Should the todo item be included in the list of todos?
217 \newif\if@todonotes@appendtolistoftodos%
218 \define@key{todonotes}{list}[]{\@todonotes@appendtolistoftodostrue}%
219 \define@key{todonotes}{nolist}[]{\@todonotes@appendtolistoftodosfalse}%

Should the todo item be displayed inline?
220 \newif\if@todonotes@inlinenote%
221 \define@key{todonotes}{inline}[]{\@todonotes@inlinenotetrue}%
222 \define@key{todonotes}{noinline}[]{\@todonotes@inlinenotefalse}%

223 \newif\if@todonotes@prependcaption%
224 \define@key{todonotes}{prepend}[]{\@todonotes@prependcaptiontrue}%
225 \define@key{todonotes}{noprepend}[]{\@todonotes@prependcaptionfalse}%

Should the note in the margin be connected to the insertion point in the text?
226 \newif\if@todonotes@line%
227 \define@key{todonotes}{line}[]{\@todonotes@linetrue}%
228 \define@key{todonotes}{noline}[]{\@todonotes@linefalse}%

Should the connection between note and insertion point be drawn in a fancy way?
(does only work if line=true)

```

```

229 \newif\if@todonotes@fancyline\@todonotes@fancylinefalse%
230 \define@key{todonotes}{fancyline}[]{\@todonotes@fancylinetrue}%
231 \define@key{todonotes}{nofancyline}[]{\@todonotes@fancylinefalse}%

Author option.
232 \newcommand{\@todonotes@author}{}%
233 \newif\if@todonotes@authorgiven%
234 \define@key{todonotes}{author}{%
235     \renewcommand{\@todonotes@author}{#1}%
236     \@todonotes@authorgiventrue}%
237 \define@key{todonotes}{noauthor}[]{\@todonotes@authorgivenfalse}%

Should the text in the list of todos be different from the text in the todonote?
238 \newcommand{\@todonotes@caption}{}%
239 \newif\if@todonotes@captiongiven%
240 \define@key{todonotes}{caption}{%
241     {figwidth}{\renewcommand{\@todonotes@caption}{#1}%
242     \@todonotes@captiongiventrue}%
243 \define@key{todonotes}{nocaption}[]{\@todonotes@captiongivenfalse}%

Change the current figure width and height.
244 \newcommand{\@todonotes@currentfigwidth}{\@todonotes@figwidth}
245 \define@key{todonotes}{%
246     {figwidth}{\renewcommand{\@todonotes@currentfigwidth}{#1-2pt}}
247 \newcommand{\@todonotes@currentfigheight}{\@todonotes@figheight}
248 \define@key{todonotes}{%
249     {figheight}{\renewcommand{\@todonotes@currentfigheight}{#1-2pt}}
250 \newcommand{\@todonotes@currentfigcolor}{\@todonotes@figcolor}
251 \define@key{todonotes}{%
252     {figcolor}{\renewcommand{\@todonotes@currentfigcolor}{#1}}

Change the width of an inline note.
253 \newcommand{\@todonotes@inlinewidth}{\linewidth}%
254 \define@key{todonotes}{%
255     {inlinewidth}{\renewcommand{\@todonotes@inlinewidth}{#1}}

Change if inline note is written to a separate line or not.
256 \newif\if@todonotes@inlinepar
257 \@todonotes@inlinepartrue
258 \define@key{todonotes}{inlinepar}[]{\@todonotes@inlinepartrue}%
259 \define@key{todonotes}{noinlinepar}[]{\@todonotes@inlineparfalse}%

Preset values of the options
260 \presetkeys%
261     {todonotes}%
262     {linecolor=\@todonotes@linecolor,%
263     backgroundcolor=\@todonotes@backgroundcolor,%
264     textcolor=\@todonotes@textcolor,%
265     bordercolor=\@todonotes@bordercolor,%
266     tickmarkheight=\@todonotes@defaulttickmarkheight,%
267     nofancyline,%
268     nodisable,%

```

```

269     noline,%
270     nocaption,%
271     noauthor,%
272     \@todonotes@shadowenabledbydefault,%
273     figwidth=\@todonotes@figwidth,%
274     figheight=\@todonotes@figheight,%
275     figcolor=\@todonotes@figcolor,%
276     line, list,%
277     inlinewidth=\linewidth,
278     inlinepar}{}%
279 \@temptokena\expandafter{\@todonotes@textsize}
280 \edef\next{\noexpand\presetkeys{todonotes}{size=\the\@temptokena}{}}
281 \next

```

2.3 The main code part

Here is the actual macros defined. If the option "disable" was passed to the package define empty commands.

```

282 \if@todonotes@disabled%
283     \newcommand{\listoftodos}[1] [] {}
284     \newcommand{\@todo}[2] [] {}
285     \newcommand{\missingfigure}[2] [] {}
286 \else % \if@todonotes@disabled

```

Define the `\listoftodos` command and define the appearance of the list of todos.

```

287 \newcommand{\listoftodos}[1] [\@todonotes@todolistname]
288     {\@ifundefined{chapter}{\section*{#1}}{\chapter*{#1}} \@starttoc{tdo}}
289 \newcommand{\l@todo}
290     {\@dottedtocline{1}{0em}{2.3em}}

```

Define styles used by the todo command

```

291 \tikzstyle{notestylera} = [
292     draw=\@todonotes@currentbordercolor,
293     fill=\@todonotes@currentbackgroundcolor,
294     text=\@todonotes@currenttextcolor,
295     line width=0.5pt,
296     text width = \@todonotes@textwidth - 1.6 ex - 1pt,
297     inner sep = 0.8 ex,
298     rounded corners=4pt]

```

`\@todo` Define the `\@todo` command.

```

299 \newcommand{\@todo}[2] [] {}%

```

Use the global value for determining the default prepend behavior.

```

300 \if@todonotes@prependcaptionglobal%
301 \@todonotes@prependcaptiontrue%
302 \else%
303 \@todonotes@prependcaptionfalse%
304 \fi%

```

Store the original text for later usage and parse the given options.

```
305 \renewcommand{\@todonotes@text}{#2}%
306 \renewcommand{\@todonotes@caption}{#2}%
307 \setkeys{todonotes}{#1}%
```

Add shadows to the inserted todonotes.

```
308 \if@todonotes@usesshadow%
309 \if@todonotes@shadowlibraryloaded%
310 \tikzstyle{notestyle} = [notestyloraw,%
311     general shadow={shadow xshift=0.5ex, shadow yshift=-0.5ex,%
312         opacity=1,fill=black!50}]%
313 \else%
314 \PackageWarning{todonotes}{Trying to put a shadow below a todonote,%
315 but the loadshadowlibrary option was not given when loading%
316 the todonotes package}%
317 \tikzstyle{notestyle} = [notestyloraw]%
318 \fi%
319 \else%
320 \tikzstyle{notestyle} = [notestyloraw]%
321 \fi%
```

Update notestyles

```
322 \tikzstyle{notestyleleft} = [%
323     notestyle,%
324     left]%
325 \tikzstyle{connectstyle} = [%
326     thick,%
327     draw=\@todonotes@currentlinecolor]%
328 \tikzstyle{inlinenotestyle} = [%
329     notestyle,%
330     text width=\@todonotes@inlinewidth - 1.6 ex - 1 pt]%
```

If the option `disable` is given to the command, no output is generated.

```
331 \if@todonotes@localdisable%
332 \else%
```

Add the item to the list of todos. When the option `colorinlistoftodos` is given to the package a small colored square is added in front of the text.

```
333 \addtocounter{@todonotes@numberoftodonotes}{1}%
334 \if@todonotes@appendtolistoftodos%
335     \phantomsection%
336     \if@todonotes@captiongiven%
337     \else%
338         \renewcommand{\@todonotes@caption}{#2}%
339     \fi%
340     \@todonotes@addElementToListOfTodos%
341 \fi%
```

Prepend the short caption given if it is requested

```
342 \if@todonotes@captiongiven%
343     \if@todonotes@prependcaption%
```

```

344     \renewcommand{\@todonotes@text}{\@todonotes@caption: #2}%
345     \fi%
346 \fi%

```

Place the todonote as indicated by the options (inline or in a marginpar), below is the code for the inline placement.

```

347 \if@todonotes@inlinenote%
348     \@todonotes@drawInlineNote%
349 \else%
350     \@todonotes@drawMarginNoteWithLine%
351 \fi%\if@todonotes@inlinenote
352 \fi%\if@todonotes@localdisable
353 }%

```

drawMarginNoteWithLine Define helper function `drawMarginNoteWithLine`.

```

354 \newcommand{\@todonotes@drawMarginNoteWithLine}{%

```

When the todonote should be placed inside a marginpar, the code below is applied. First is the current location in the document stored, this enables us later to connect this point with the inserted todonote.

```

355 \ifvmode
356     \vspace*{-\parskip}%    % backup if we are already in vertical mode
357     \vskip-\baselineskip    % (and don't lose that space after a
358                             % pagebreak ...
359     \noindent
360 \fi
361 \begin{tikzpicture}[remember picture, overlay, baseline=-0.75ex]%
362     \node [coordinate] (inText) {};%
363 \end{tikzpicture}%
364 \marginpar[{} Draw note in left margin
365     \@todonotes@drawMarginNote%
366     \@todonotes@drawLineToLeftMargin%

```

In the book documentclass (which is a twoside layout), the `\marginpar` macro takes two arguments `\marginpar[left]{right}`. If both arguments are given, latex will decide in which side the margin note has to be inserted, and then use the corresponding commands.

```

367 }]{% Draw note in right margin
368     \@todonotes@drawMarginNote%
369     \@todonotes@drawLineToRightMargin%
370 }%
371 }%

```

addElementToListOfTodos Define helper function `addElementToListOfTodos`.

```

372 \newcommand{\@todonotes@addElementToListOfTodos}{%
373     \if@todonotes@colorinlistoftodos%
374         \addcontentsline{tdo}{todo}{%
375             \fcolorbox{\@todonotes@currentbordercolor}%
376                 {\@todonotes@currentbackgroundcolor}%
377                 {\textcolor{\@todonotes@currentbackgroundcolor}{o}}%

```

```

378         \ \@todonotes@caption}%
379     \else%
380         \addcontentsline{tdo}{todo}{\@todonotes@caption}%
381     \fi}%

```

`drawInlineNote` Define helper function `useSizeCommand`.

```

382 \newcommand{\@todonotes@useSizeCommand}{%
383 \ifcsname \expandafter\string\@todonotes@sizecommand\endcsname
384 \csname \expandafter\string\@todonotes@sizecommand\endcsname%
385 \else
386 \@todonotes@sizecommand
387 \fi%
388 }%

```

`drawInlineNote` Define helper function `drawInlineNote`.

```

389 \newcommand{\@todonotes@drawInlineNote}{%
390     \if@todonotes@dviStyle%
391         {\if@todonotes@inlinepar\par\noindent\fi%
392         \begin{tikzpicture}[remember picture]%
393             \draw node[inlinenotestyle] {};
394         \end{tikzpicture}%
395         \if@todonotes@inlinepar\par\fi}%
396     \if@todonotes@authorgiven%
397         {\noindent \@todonotes@useSizeCommand \@todonotes@author:\,\@todonotes@text}%
398     \else%
399         {\noindent \@todonotes@useSizeCommand \@todonotes@text}%
400     \fi
401     {\if@todonotes@inlinepar\par\noindent\fi%
402     \begin{tikzpicture}[remember picture]%
403         \draw node[inlinenotestyle] {};
404     \end{tikzpicture}%
405     \if@todonotes@inlinepar\par\fi}%
406 \else%
407     {\if@todonotes@inlinepar\par\noindent\fi%
408     \begin{tikzpicture}[remember picture]%
409         \draw node[inlinenotestyle,font=\@todonotes@useSizeCommand] {%
410             \if@todonotes@authorgiven%
411                 {\noindent \@todonotes@author:\,\@todonotes@text}%
412             \else%
413                 {\noindent \@todonotes@text}%
414             \fi};%
415     \end{tikzpicture}%
416     \if@todonotes@inlinepar\par\fi}%
417 \fi}%

```

`drawMarginNote` Define helper function `drawMarginNote`.

```

418 \newcommand{\@todonotes@drawMarginNote}{%
419 \if@todonotes@dviStyle%
420     \begin{tikzpicture}[remember picture]%

```

```

421     \draw node[notestyle] {};%
422 \end{tikzpicture}\}%
423 \begin{minipage}{\@todonotes@textwidth}%
424 \if@todonotes@authorgiven%
425     \@todonotes@useSizeCommand \@todonotes@author \@todonotes@text%
426 \else%
427     \@todonotes@useSizeCommand \@todonotes@text%
428 \fi%
429 \end{minipage}\}%
430 \begin{tikzpicture}[remember picture]%
431     \draw node[notestyle] (inNote) {};%
432 \end{tikzpicture}%
433 \else%
434     \let\originalHbadness\hbadness%
435     \hbadness 100000%
436     \begin{tikzpicture}[remember picture,baseline=(X.base)]%
437         \node(X){\vphantom{\@todonotes@useSizeCommand X}};%
438         \if@todonotes@authorgiven%
439             \draw node[notestyle,font=\@todonotes@useSizeCommand,anchor=north] (inNote) at (X.n
440                 {\@todonotes@author};%
441             \node(Y)[below=of X]{};%
442             \draw node[notestyle,font=\@todonotes@useSizeCommand,anchor=north] (inNote) at (X.s
443                 {\@todonotes@text};%
444         \else%
445             \draw node[notestyle,font=\@todonotes@useSizeCommand,anchor=north] (inNote) at (X.n
446                 {\@todonotes@text};%
447         \fi%
448     \end{tikzpicture}%
449     \hbadness \originalHbadness%
450 \fi}%

```

`drawLineToRightMargin` Define helper function `drawLineToRightMargin`.

```

451 \newcommand{\@todonotes@drawLineToRightMargin}{%
452 \if@todonotes@line%
453 \if@todonotes@fancyline%
454 \tikz[remember picture,overlay]{%
455 \tikzstyle{both}=[line width=3pt, draw, opacity=0.15]%
456 \tikzstyle{line}=[shorten >=5pt, line cap=round]%
457 \tikzstyle{head}=[shorten >=-1pt, dash pattern=on 0pt off 1pt, ->%
458 \foreach \s in {line,head}{%
459 \draw[both,\s]%
460 (inNote.north west).. controls +(0:0) and +(90:1.5)..([yshift=1ex] inText);%
461 }%
462 }%
463 \else%
464 \begin{tikzpicture}[remember picture, overlay]%
465 \draw[connectstyle]%
466 ([yshift=-0.2cm + \@todonotes@tickmarkheight] inText)%
467 -| ([yshift=-0.2cm] inText)%
468 -| ([xshift=-0.2cm] inNote.west)%

```

```

469 -| (inNote.west);%
470 \end{tikzpicture}%
471 \fi%
472 \fi}%

```

`drawLineToLeftMargin` Define helper function `drawLineToLeftMargin`.

```

473 \newcommand{\@todonotes@drawLineToLeftMargin}{%
474 \if@todonotes@line%
475 \if@todonotes@fancyline%
476 \tikz[remember picture,overlay]{%
477 \tikzstyle{both}=[line width=3pt, draw, opacity=0.15]%
478 \tikzstyle{line}=[shorten >=5pt, line cap=round]%
479 \tikzstyle{head}=[shorten >=-1pt, dash pattern=on 0pt off 1pt,->%
480 \foreach \s in {line,head}{%
481 \draw[both,\s]%
482 (inNote.north east).. controls +(0:0) and +(90:1.5)..([yshift=1ex] inText);%
483 }%
484 }%
485 \else%
486 \begin{tikzpicture}[remember picture, overlay]%
487 \draw[connectstyle]%
488 ([yshift=-0.2cm + \@todonotes@tickmarkheight] inText)%
489 -| ([yshift=-0.2cm] inText)%
490 -| ([xshift=0.2cm] inNote.east)%
491 -| (inNote.east);%
492 \end{tikzpicture}%
493 \fi%
494 \fi}%

```

`\missingfigure` Defines the `\missingfigure` macro.

```

495 \newcommand{\missingfigure}[2] []{%
496 \setkeys{todonotes}{#1}%
497 \addcontentsline{tdo}{todo}{\@todonotes@MissingFigureText: #2}%
498 \par
499 \noindent
500 \hfill
501 \begin{tikzpicture}
502 \draw[fill=\@todonotes@currentfigcolor, draw = black!40, line width=2pt]
503   (-2, -0.5*\@todonotes@currentfigheight-0.5cm)
504   rectangle +(\@todonotes@currentfigwidth, \@todonotes@currentfigheight);
505 \draw (2, -0.5) node[right, text
506   width=\@todonotes@currentfigwidth-4.5cm, font=\@todonotes@useSizeCommand] {#2};
507 \draw[red, fill=white, rounded corners = 5pt, line width=10pt]
508   (30:2cm) -- (150:2cm) -- (270:2cm) -- cycle;
509 \draw (0, 0.3) node {\@todonotes@MissingFigureUp};
510 \draw (0, -0.3) node {\@todonotes@MissingFigureDown};
511 \end{tikzpicture}\hfill
512 \null\par
513 }% Ending \missingfigure command
514 \fi% Ending \@todonotes@ifdisabled

```


`\todotoc` Inserts a reference to the list of todos in the table of contents. If `chapter` is defined, `chapter` is used as level otherwise will `section` be used. The `\todotoc` command respects the `disable` option.

```

515 \newcommand{\todotoc}
516 {%
517     \if@todonotes@disabled
518     \else
519 \addcontentsline{toc}{\@ifundefined{chapter}{section}{chapter}}{\@todonotes@todolistname}%
520     \fi
521 }

```

`\todo` Define the `\todo` command as a redirection to `\@todo`.

```

522 \newcommand{\todo}[2] []{%
523 % Needed to output any dangling \item of a noskip section (see #36):
524 \if@inlabel \leavevmode \fi
525 \if@noskipsec \leavevmode \fi
526 \if@todonotes@inlinepar
527     \ifhmode
528         \@bsphack
529         \@todonotes@vmodedefalse
530     \else
531         \@savsf\@m
532         \@savsk\z@
533         \@todonotes@vmodetrue
534     \fi
535     {\@todo[#1]{#2}}%
536     \@esphack%
537     \if@todonotes@vmode \par \fi
538 \else%
539     \@todo[#1]{#2}%
540 \fi}
541 \newif\if@todonotes@vmode

```

Change History

0.1	General: The first version of the package	1	0.4.2	General: Fixed a bug with the disable option to the package. . .	1
0.2	General: Updated the option handling of the package	1	0.5	General: Created a dtx file containing both source code and documentation of the package	1
0.2.1	General: Slightly modified by Kjell Magne Fauske to support notes in the left margin (for documentstyle book).	1	0.5.1	General: Updated the documentation	1
0.2.2	General: Added a missingfigure command	1	0.5.2	General: Fixed a bug that prevented the usage of the option french for babel. Bug report by Thomas Braun.	1
0.2.3	General: Made a dependency on the calc package	1	0.6	General: Added the caption option to the todo command.	1
0.3	General: Delayed the requirements for the hyperref package untill begin document and added an optional argument to the todo command for adding inline todonotes (Idea from Patrick Toche)	1	0.6.1	General: Added a new usecase with decreased line spacing.	1
0.3.1	General: Added some options to the todo macro (Idea: Patrick Toche) and made the listoftodos point at the inserted todos and not only the current / previous section, subsection or figure using the phantomsection macro.	1	0.6.2	General: Added a usecase by Fabrice Niessen.	1
0.4	General: Modified the behaviour of the inline todonotes, to avoid empty lines around the inline todonotes.	1	0.7	General: Added language options on request from Peter Zimmermann.	1
0.4.1	General: Added the option colorinlistoftodos which inserts a small box with the used fillcolor of the todonotes in the list of todos.	1	0.7.1	General: Reworked the color options for both the whole package and the todo command. General code clean up. Added the prependcaption package option.	1
			0.7.2	General: Avoid to change the fontsize inside the list of todos, fixing a bug revealed by Vladimir Zhuravlev.	1
			0.7.3	General: The localization options (danish and german) and the disable options, were all flawed by naming inconsistencies that	

	made then break the package.	1	0.8.4	
0.7.4	This have been fixed.	1		General: Fixed a bug related to the obeyDraft option. 1
	General: Fixed a bug related to the caption option for the todo command. Introduced a counter of the number of inserted todos.	1	0.8.5	General: Added two new usecases (enumeration of inserted todonotes and how to set custom default values). Changed the order of the use case examples. 1
0.7.5	General: Fixed a typo in a macroname.	1	0.8.6	General: Added a portuguese translation by Og DeSouza. . . . 1
0.7.6	General: Added a textsize option for the package and the prepend / noprepnd option for the todo command.	1	0.8.7	General: Updated portuguese translation. Added a ngerman alias for the german translation suggested by Michael Niedermair. 1
0.8	General: Added three new translations french, spanish and catalan thanks to Richard Dominique and Joan Queralt. Improved the visual appearence of the inserted notes (rounded corners and optional shadows) with code from Joan Queralt. Found an untranslated textstring "Figure" in the source. Added a figwidth option to the missingfigure command, patch by Paul Ivanov.	1	0.8.8	General: Added a new usecase from Vitaly. Fixed a bug reported by Oscar Gustafsson. Explained why the placement of todonotes in the margin fails in certain custom document classes. 1
0.8.1	General: Added a space between the colored square and the text in the list of todos. Added a new usecase for marking old / new sections. Made the name of listoftodos changeable.	1	0.8.9	General: Added a dutch translation by Ruben Ruben Vermeersch. 1
0.8.2	General: Italian translation by Gustavo Cevolani. Removed the dependence on the hyperref package.	1	0.9.0	General: Added a english option as suggested by Marco Berghoff. . . 1
0.8.3	General: Added a use case for linking to the list of todos, idea from Andreas Plank. Introduced a package option for listening to the draft option given to the document class. . . . 1	1	0.9.1	General: Added the todototoc command by idea from Sven Augustin. 1
			0.9.2	General: Use chapter (if available) for the list of todos heading. . . . 1
			0.9.3	General: Make an internal definition of the todo command, for easing redefinition of the command behaviour. 1
			0.9.4	General: Make the disable option work on a local scale. 1

0.9.5	General: Code simplification by extracting functionality to smaller macros.	1	using the presetkeys command. Updated list of alternatives to the todonotes package. Draw borders around coloured boxes in the list of todos, patch by Ze Loff.	1
0.9.6	General: Give fontsize to TikZ. Align notes with line where note is set. Added new option fancyline. Patches by Benjamin Kellermann.	1	1.0.3	General: Added the option figcolor to missing figure, patch by Pascal Hebbeker. Added Croatian translation by Ican Kokan. Changed default with of missingfigure. Removed some underfull box warnings, solution by Ernst Blecha.
0.9.7	General: Updated documentation.	1	1.0.4	General: Restructured documentation and placed some examples in the doc/examples subdirectory.
0.9.8	General: Suppress warnings about underfull / overfull boxes generated by the inserted todonotes. Patch by Peter M Schuler.	1	1.0.5	General: Example of how to change the space eating behaviour added by Anselm Wagner. Reduced the width of missingfigure to avoid bad box warnings.
0.9.9	General: Added author option, implementation provided by Xavier Alameda-Pineda. Example of modifying the listoftodos removing some protect commands with no effect.	1	1.0.6	General: Changed how whitespaces are handled around inserted todos to better mimick macros like index (Suggestion by Frank Mittelbach). Fix missing character warnings using patch by Niels Anders Danielsson. Use the specified textsize for missingfigure, patch by Johannes Twittmann. Added new options for inline todonotes based on code from Daniel Krenn. Added warning message when marginparwidth is less than 2cm, as suggested by ErikBoesen.
1.0.0	General: Mention trouble with the classicthesis style. Refer to some alternatives to the package. Added todoin command as suggested by Stefan Pinnow. Described how to use tikz externalize with todonotes. Added obeyDraft and obeyFinal options.	1	1.0.7	General: Changed the warning about marginparwidth to a proper warning message.
1.0.1	General: Fix spacing issues reported by Jonathan Zachhuber and Brent Longborough. Added figheight option to the missingfigure command as suggested by Kim Albertsson.	1	1.0.8	General: Improved spacing
1.0.2	General: Added Swedish translation by Emil Lundberg. Added usecase by Tobias Winchen. Mentioned that default arguments can be set			

behaviour so it mimics the index macro, patch by Frank Mittelbach. Center graphics vertically in missingfigure, patch by Philipp Allgeuer. Added new option tickmarkheight as suggested by Richard Niland.	1	noshadow options for the todo command.	1
1.1.0		1.1.2	
General: Many improvements to the user interface by Frank Mittelbach. Added the command <code>setuptodonotes</code> . Added <code>textcolor</code> as an option. Marked the insertion point of notes with tickmarks with a customizable height. Steamline which options that are accepted to size commands (both commands (<code>backslash tiny</code>) and <code>text (tiny)</code> are now supported).	1	General: Fix issue 36 and 37.	1
1.1.1		1.1.3	
General: Implemented shadow and		General: Fix issue 48 and make colors used in the last todo globally available.	1
		1.1.4	
		General: Fix issue 51 by avoid loading the <code>tikz shadows</code> library by default.	1
		1.1.5	
		General: Fix issue 54 related to additional spacing around todos with shadows. Actually use the <code>tickmarkheight</code> option as suggested by José Francisco Loff library by default. Described a workaround for using the standalone document class.	1