The grabbox package

Jonathan P. Spratte[†]

Version 1.4
Released 2019-05-08

Contents

1 Introduction

Sometimes I happen to write macros and environments which don't care for the exact contents of an argument but only for that contents' typeset representation and its dimensions. In that case I personally dislike the fact that those arguments couldn't contain verbatim material if coded straight forward for macros. For environments this is quite easy to create thanks to `lrbox`, for macros this approach unfortunately doesn't work without the enduser's cooperation. Thus the macros distributed hereby came into existence.

This package provides `\grabbox` to grab an argument inside of a box. The used mechanism allows category code changes in that argument as long as it is used in a place allowing category code changes (so not inside of another argument).

It is written as a docstrip file: executing `latex grabbox.dtx` generates the `grabbox.sty` file and typesets this documentation; execute `tex grabbox.dtx` to only generate `grabbox.sty`.

2 Acknowledgement

I want to thank Enrico Gregorio for helping me develop first versions of the used mechanisms for the second iteration of my `ducksay` package. If he hadn't helped me back then, I wouldn't have considered the used method further – because the user interface would've been too clumsy and require strange markup like `\foo arg}` – and therefore this package wouldn't have been created.

Additionally I want to thank David Carlisle for helping me making `\grabbox` respect surrounding text colours.

[†] E-mail: jspratte@yahoo.de

3 The macro

`\grabbox`

`\grabbox`⟨***⟩`[`⟨*inject pre pre*⟩`]{`⟨*box register*⟩`}[`⟨*inject pre*⟩`]{`⟨*box type*⟩`}[`⟨*inject post*⟩`]{`⟨*afterwards*⟩`}`

grabs the next braced argument and stores it inside of the box ⟨*box register*⟩. The box is of ⟨*box type*⟩, which should be one of `\hbox` or `\vbox` or `\vtop`. The contents of the box except for ⟨*inject pre pre*⟩ and ⟨*inject post*⟩ will be contained in an additional level of grouping to ensure colour safety (similar to LaTeX's `\sbox`). ⟨*inject pre pre*⟩ will be injected at the beginning of the box before this additional group is opened, ⟨*inject pre*⟩ will be injected at the beginning of the box and can affect its contents, ⟨*inject post*⟩ will be injected at the end of the box but can't be affected by stuff inside of ⟨*inject pre*⟩ or added content unless they are using global definitions – ⟨*inject pre pre*⟩ however can affect the contents of ⟨*inject post*⟩. Unless the ⟨***⟩ is given leading and trailing spaces will be stripped from the box. After the box is read in ⟨*afterwards*⟩ will be inserted. The complete contents of the box will be something like:

⟨*inject pre pre*⟩`{\set@color`⟨*inject pre*⟩⟨*argument*⟩`}`⟨*inject post*⟩

All assignments are made local. Currently it is not safe to nest macros which use `\grabbox`. It should become safe if your macros use `\grabbox` inside of a group, so the inner `\grabbox` doesn't affect the outer one.

`\grabbox` uses `\afterassignment` and `\aftergroup` to do its magic. The former should be safe where it is used, the latter is used inside of the boxed argument before any contents are inserted.

Since `\grabbox` works by setting a boxregister using `\setbox` (and a bunch of temporary macros), it is of course not expandable and defined `\protected`.

`\@grabbox`

`\@grabbox`⟨***⟩`{`⟨*inject pre pre*⟩`}{`⟨*box register*⟩`}{`⟨*inject pre*⟩`}{`⟨*box type*⟩`}{`⟨*inject post*⟩`}{`⟨*afterwards*⟩`}`

This is a variant of `\grabbox` that should be faster because it doesn't parse for optional arguments. Instead every argument is mandatory except for the star, just leave the arguments empty if you'd otherwise not use the corresponding optional argument in `\grabbox`.

4 Useless Example!

First we need to reserve us a box register for this example:

```
\newsavebox\ourbox
```

Next we define a macro which takes some arguments and uses `\grabbox`:

```
\newcommand\examplecmd[2]
  {%
    \begingroup
    \grabbox\ourbox[\itshape]\hbox[ \sffamily is]{\examplecmdOut{#1}{#2}}
  }
```

And we need our helper macro which is executed after `\grabbox`:

```
\newcommand\examplecmdOut[3]
  {%
    \begin{tabular}[t]{@{}ll@{}}
      Arg1: & #1\\
      Arg2: & #2\\
      Box:  & \unhbox\ourbox\\
      Arg3: & #3
    \end{tabular}%
    \endgroup
  }
```

The result is a macro that takes two ordinary arguments, after those a box in horizontal mode and finally another ordinary argument. If we use this macro we get the following:

Arg1:	Hi,
Arg2:	my
Box:	*\name* is
Arg3:	Steve!

One can see that `\sffamily is` in ⟨*`inject post`*⟩ is not affected by the `\itshape` in ⟨*`inject pre`*⟩. The used code to generate that table was:

```
\examplecmd{Hi,}{my}{\verb|\name|}{Steve!}
```

5 Useful Example?

This example provides a macro which typesets its mandatory argument in a block of a definable number of lines, it is meant for a single paragraph.

```
% Getting a box register:
\newsavebox\RectangleBox
% Defining the main macro:
\newcommand\Rectangle[1][4]
  {%
    \begingroup
    \grabbox\RectangleBox\hbox
      {%
        % Since we don’t want to read more arguments after the box,
        % we don’t need a second macro and can put the output routine
        % here.
        \begin{minipage}{\dimexpr\wd\RectangleBox/#1\relax}
          \parfillskip0pt
          \unhbox\RectangleBox
        \end{minipage}%
        \endgroup
      }%
  }
```

As you can see, this macro uses `\grabbox` in a group delimited by `\begingroup` and `\endgroup` – like the useless example. It should therefore be safe to nest it inside other macros using `\grabbox`.

Finally a usage example of our new macro (with the duckuments package loaded):

```
\begin{center}
  \Rectangle[9]{\blindduck}
\end{center}
```

Results in:

> There once was a very smart but sadly blind duck. When it was still a small duckling it was renowned for its good vision. But sadly as the duck grew older it caught a sickness which caused its eyesight to worsen. It became so bad, that the duck couldn't read the notes it once took containing much of inline math. Only displayed equations remained legible. That annoyed the smart duck, as it wasn't able to do its research any longer. It called for its underduckling and said: 'Go, find me the best eye ducktor there is. He shall heal me from my disease!'

6 Implementation

```
⟨*pkg⟩
\@ifdefinable{\if@grabbox@spaces@}{\newif\if@grabbox@spaces@}
\@ifdefinable{\grabbox@def}
  {\long\def\grabbox@def#1#2#{\grabbox@def@a{}#1{#2}}}
\@ifdefinable{\grabbox@ldef}
  {\long\def\grabbox@ldef#1#2#{\grabbox@def@a\long#1{#2}}}
\@ifdefinable{\grabbox@def@a}
  {%
    \protected\long\def\grabbox@def@a#1#2#3#4%
      {\@ifdefinable#2{\protected#1\def#2#3{#4}}}%
  }
\newcommand\grabbox@def@step[4]
  {%
    \grabbox@def#1##1{\def#2{##1}\grabbox@opt#3#4}%
  }
\grabbox@def\grabbox@opt#1#2%
  {%
    \@ifnextchar[
      {\grabbox@opt@get#1#2}
      {\def#1{}#2}%
  }
\grabbox@ldef\grabbox@opt@get#1#2[#3]%
  {%
    \def#1{#3}#2%
  }
\grabbox@def\grabbox@set@color
  {%
    \@ifundefined{set@color}{}
      {\global\let\grabbox@set@color\set@color\grabbox@set@color}%
  }
```

```
\AtBeginDocument
  {%
    \@ifundefined{set@color}
      {\gdef\grabbox@set@color{}}
      {\global\let\grabbox@set@color\set@color}%
  }%
\newcommand*\grabbox@unskip@space
  {%
    \ifhmode\unskip\fi
  }
\grabbox@def\grabbox
  {%
    \@ifstar
      {\@grabbox@spaces@true\grabbox@a}
      {\@grabbox@spaces@false\grabbox@a}%
  }
\grabbox@def\grabbox@a
  {%
    \grabbox@opt\grabbox@into@prepre\grabbox@b
  }
\grabbox@def@step\grabbox@b\grabbox@name\grabbox@into@pre\grabbox@c
\grabbox@def@step\grabbox@c\grabbox@type\grabbox@into@post\grabbox@d
\grabbox@ldef\grabbox@d#1%
  {%
    \def\grabbox@final{#1}%
    \afterassignment\grabbox@intermediate
    \setbox\grabbox@name\grabbox@type
  }
\grabbox@def\@grabbox
  {%
    \@ifstar
      {\@grabbox@spaces@true\@grabbox@a}
      {\@grabbox@spaces@false\@grabbox@a}%
  }
\grabbox@ldef\@grabbox@a#1#2#3#4#5%
  {%
    \def\grabbox@into@prepre{#1}%
    \def\grabbox@name       {#2}%
    \def\grabbox@into@pre   {#3}%
    \def\grabbox@type       {#4}%
    \def\grabbox@into@post  {#5}%
    \grabbox@d
  }
\grabbox@def\grabbox@intermediate
  {%
    \grabbox@into@prepre
    \bgroup
    \if@grabbox@spaces@
    \else
      \aftergroup\grabbox@unskip@space
    \fi
    \grabbox@set@color
    \aftergroup\grabbox@after
    \grabbox@into@pre
    \if@grabbox@spaces@
    \else
      \ignorespaces
    \fi
  }
\newcommand*\grabbox@after@aux@b[1]
  {%
    \grabbox@after@aux@a
  }
\grabbox@def\grabbox@after@aux@a
  {%
    \@ifnextchar\reset@color
      {\reset@color\grabbox@after@aux@b}
      {\egroup\grabbox@final}%
  }
\grabbox@def\grabbox@after
  {%
    \grabbox@into@post
    \endgraf
    \grabbox@after@aux@a
  }
```

```
\endinput
```

⟨/pkg⟩