

---

## The DuckBoat — Beginners' Pond: Tcolorchat!

Herr Professor Paulinho van Duck

### Abstract

In this installment, Prof. van Duck will show you how to typeset chat-like conversations using `tcolorbox`, a package for creating customized text boxes.

### 1 Wunderschöne Tage in Bonn!

Hi, (L)TeX friends!

Last July, for the first time, my assistant Carla and I attended a TUG meeting in person. We spent some exciting days in Bonn, Germany, listening to captivating talks, in an astonishing location, a former monastery.

We met a lot of friends from countries all around the world, Europe, of course, but also the USA, Brazil, and India! We joyfully gave a face to many nicknames.

Our local guest, Gert Fischer, took us on a fascinating city tour. We found out interesting things about Bonn, such as that the firm famous for inventing gummi bears and licorice wheels is based there! We thank him, Ulrike Fischer, and the other members of the conference committee for the perfect organization.

We dined at a skyscraper restaurant with a gorgeous view of the Rhine Valley. We took a river trip to Königswinter, enjoyed the view from the hill, and visited the ruins of Drachenfels Castle. We also stopped at Schloss Drachenburg, technically not a castle, but a nineteenth-century mansion.

We had a lot of fun and, last but not least, David Carlisle did not eat me,<sup>1</sup> quack!



Talking with some TeXnicians at the meeting, I found out that the `tabularray` package has a flaw I should have mentioned in my previous article [4].

Unfortunately, `tabularray` significantly slows down the compilation of your document. If you have more than a dozen tables (the exact number depends on their complexity), it is better to use the traditional environments, possibly reserving `tabularray` for the few out-of-the-ordinary tables.

It is a pity that such a gorgeous package has this defect, I hope it will be somehow fixed in the future.




---

<sup>1</sup> For the ones who do not usually attend TeX.SE chat: see Box 1 and footnote 3.

In this installment, I would like to show you some features of `tcolorbox`, a convenient package to typeset colored and framed text boxes. The idea comes from the post *Chat bubbles with picture and name attached*<sup>2</sup> on TeX.SE.

The dialogues in the examples (except for Box 1) are taken from famous movies. Try to guess them without searching on the internet, quack!

Eventually, I would like to thank the user *frougon* from TopAnswers TeX ([topanswers.xyz/tex](https://topanswers.xyz/tex)) for their precious help with the examples in Boxes 5 and 6; as well as *samcarter*, Gert Fischer, and Enrico Gregorio for their proofreading and suggestions.

### 2 Quack Guide n. 9:

#### Chat-like messages with `tcolorbox`

The package `tcolorbox` is very versatile, allowing creation of text boxes with a lot of features. It also has a detailed manual [6], rich in useful examples. I strongly advise you to look at its Section 2 *Quack Reference*, where some of the main options are visualized.

For the sake of brevity, I will not illustrate all the possible customizations you can apply to your boxes. I will only try to create something that resembles a chat dialogue, proceeding step by step, adding a few options at a time to get the desired result.

The main environment of the package is

```
\begin{tcolorbox}[<options>]
<environment content>
\end{tcolorbox}
```

In the optional argument, you can list all the specific *<options>* for that box.

The *<environment content>* can be divided into a mandatory upper part and an optional lower part. In this installment, I will use `tcolorboxes` with the upper part only.

There is also a command

```
\tcbox[<options>]{<box content>}
```

that creates a colored box fitted to the width of *<box content>*. The main differences from the `tcolorbox` environment are that it cannot have a lower part and cannot be broken across pages.



In some of the following examples, you will only find the `tcolorbox` environments and the possible settings needed for them.

Remember to first load the package in your preamble:

```
\usepackage{tcolorbox}
```

---

<sup>2</sup> [tex.stackexchange.com/questions/624775/chat-bubbles-with-picture-and-name-attached](https://tex.stackexchange.com/questions/624775/chat-bubbles-with-picture-and-name-attached)

## 2.1 The standard

Standard `tcolorboxes` appear as black and white boxes with rounded corners.

You can also add a heading line with the option `title={⟨text⟩}`; see Box 1.<sup>3</sup>

**Box 1 – Standard `tcolorboxes`**

```
\begin{tcolorbox}[title={Paulo}]
  Quack!
\end{tcolorbox}
\begin{tcolorbox}[title=David]
  Dinner!
\end{tcolorbox}
```

---

Paulo

Quack!

David

Dinner!

The brackets around the title are not mandatory, but it is a good practice to add them, because in some cases, for example, if there is an equal sign or a comma in the title text, they *are* needed.

## 2.2 Aspect customization and border dimension

Let us start with the title setting, I would like to use it to show who is chatting. The option `fonttitle=⟨text⟩` puts the content of `⟨text⟩` before the title. I used it to set a bold, script-size font. I also added a little vertical space before/after the title text with `toptitle/bottomtitle=⟨length⟩`.

Since I would like to make the box more lively, I added some colors, with `coltitle=⟨color⟩` for the title text, `colbacktitle=⟨color⟩` for the background of the title, and `colback=⟨color⟩` for the background of the main text. See Box 2, and please use your imagination if you are reading this article in black and white.

Since the sent messages should appear horizontally aligned to the right (and the received ones to the left), I took advantage of the options: `halign title=⟨alignment⟩`, for the title (aligned to the left by default), and `halign=⟨alignment⟩`, for the content (initially justified).

I also do not like the border lines, so I set their dimension to zero, using `titlerule=⟨length⟩`, for the line between the title and the text; `boxrule=⟨length⟩`,

<sup>3</sup> This Box represents the usual conversation between Paulo Cereda and David Carlisle in the TeX.SE chat.

for the border lines; and `boxsep=⟨length⟩`, for the padding between the text content and the frame of the box.

As you can see in Box 2, this is not enough. Even setting their width to zero points, the rules are still visible, even if barely.

**Box 2 – Zero dimension rules still visible**

```
% In your preamble:
\tcbset{fonttitle=\bfseries\scriptsize,
  titlerule=0pt, boxrule=0pt, boxsep=0pt,
  toptitle=6pt, bottomtitle=2pt}

% In your document:
\begin{tcolorbox}[title={Leia},
  halign=flush left,
  coltitle=red, colbacktitle=yellow!30,
  colback=yellow!30]
  I love you
\end{tcolorbox}
\begin{tcolorbox}[title={Han},
  halign title=flush right,
  halign=flush right,
  coltitle=blue, colbacktitle=green!30,
  colback=green!30]
  I know
\end{tcolorbox}
```

---

Leia

I love you

Han

I know

Of course, there is a trick to avoid it, quack! The option `opacityframe=⟨fraction⟩` sets the frame opacity of the box to the given fraction. I made it transparent with `opacityframe=0`; see Box 3.



Using `\tcbset{⟨options⟩}`, you can set options for all top-level `tcolorboxes` in your document. If you would also like to set them for nested boxes, there is `\tcbsetforeverylayer{⟨options⟩}`.

If you do not want to have a general setting, you can create a `style` and use it only when needed, like `commonoptions` in Box 3. The same Box shows that you can also create customized `tcolorbox` environments, like `sentmsg` and `receivedmsg`, with the command

```
\newtcolorbox[⟨init options⟩]{⟨name⟩}
  [⟨number⟩][⟨default⟩]{⟨options⟩}
```

where `⟨name⟩` is the name of the new environment,

featuring some *⟨options⟩*, a certain *⟨number⟩* of parameters (the first one could be optional with a *⟨default⟩* value), and some *⟨init options⟩* in case the box should be numbered. See Section 3.2 *Producing tcolorbox Environments and Commands* of the package documentation [6] for details.

### Box 3 – No frame at all

```
% In your preamble:
\tcbset{
  commonoptions/.style={
    fonttitle=\bfseries\scriptsize,
    titlerule=0pt, boxrule=0pt, boxsep=0pt,
    toptitle=6pt, bottomtitle=2pt,
    opacityframe=0,
  }
}
\newtcolorbox{receivedmsg}[2] []{
  commonoptions,
  halign title=flush left,
  halign=flush left,
  coltitle=red, colbacktitle=yellow!30,
  colback=yellow!30,
  title={#2},
  #1}
\newtcolorbox{sentmsg}[2] []{
  commonoptions,
  halign title=flush right,
  halign=flush right,
  coltitle=blue, colbacktitle=green!30,
  colback=green!30,
  title={#2},
  #1}

% In your document:
\begin{receivedmsg}{Marion}
  You're not the man I knew ten years ago.
\end{receivedmsg}
\begin{sentmsg}{Indiana}
  It's not the years, honey, it's the
  mileage.
\end{sentmsg}
```

**Marion**

You're not the man I knew ten years ago.

**Indiana**

It's not the years, honey, it's the mileage.

For more complex environments, you could use `\NewTColorBox`, which is analogous to the macro `\NewDocumentEnvironment`:

```
\NewTColorBox[⟨init options⟩]{⟨name⟩}
  {⟨specification⟩}{⟨options⟩}
```

The meaning of the parameters is the same as in `\newtcolorbox` but, in addition, you can set the argument *⟨specification⟩*.

In Boxes 5 and 6, you will find some examples of environments with a starred `s`, an optional `o`, and a mandatory argument `m`. See `usrguide`, the user-mode documentation for L<sup>A</sup>T<sub>E</sub>X [2], for a description of all possible argument types.



An alternative method to get rid of the borders is loading the library `skins`:

```
\tcbuselibrary{skins}
```

and using `skin=enhanced` or just `enhanced`, which translates the drawing commands into TikZ path commands.

It also allows using the option `frame hidden`, a shortcut for

```
frame style={draw=none, fill=none}
```

You can find an example in Box 4.



Other than `skins`, `tcolorbox` has several libraries. For example, to cite only some, `listings` or `minted`, which load the homonymous packages for printing source code; `theorems`, which loads `amsmath` and provides macros for typesetting boxed theorems and similar environments; `breakable`, which allows you to automatically break your boxes from one page to another; and `fitting`, which provides support for adapting the font size for your text to the box dimensions.

It is often the case that you need more than one of them. The package makes available three shortcuts to load some or all of them at once: `many`, which loads the libraries for the box settings; `most`, which loads all libraries except `minted` and `documentation`; and `all` that loads all libraries.

For details, see Section 1.3 *Libraries* of the package manual [6].

## 2.3 Dimension and position

The chat messages are usually positioned on the right or the left, depending on if they are sent or received.

You may create this effect by taking advantage of the options `left/right skip=⟨length⟩`; see Box 4. They insert some horizontal space of the given length before/after the `tcolorbox`.

However, this is not enough to mimic the usual look of a chat. If the messages are brief, they do not enlarge to the border of your phone/computer screen. The frame should adapt to the width of the text. You can achieve this using two options together.

## Box 4 – Library skins and option skip

```
% In your preamble:
\usepackage{tcolorbox}
\tcbuselibrary{skins}
\tcbset{
  commonoptions/.style={
    enhanced,
    frame hidden,
    fonttitle=\bfseries\scriptsize,
    titlerule=0pt, boxrule=0pt, boxsep=0pt,
    toptitle=6pt,
    bottomtitle=2pt
  },
}
\newtcolorbox{receivedmsg}[2] [] {
  commonoptions,
  right skip=.7cm,
  halign title=flush left,
  halign=flush left,
  coltitle=red,
  colbacktitle=yellow!30,
  colback=yellow!30,
  title={#2},
  #1}
\newtcolorbox{sentmsg}[2] [] {
  commonoptions,
  left skip=.7cm,
  halign title=flush right,
  halign=flush right,
  coltitle=blue,
  colbacktitle=green!30,
  colback=green!30,
  title={#2},
  #1}

% In your document:
\begin{receivedmsg}{Ilsa}
  But what about us?
\end{receivedmsg}
\begin{sentmsg}{Rick}
  We'll always have Paris.
\end{sentmsg}
```

**Ilsa**

But what about us?

**Rick**

We'll always have Paris.

The first is `hbox`, a shortcut for `capture=hbox`; it sizes the box according to its content. It is the default for `\tcbbox`. It does not allow boxes with a lower part or with text split into more rows.

The latter limitation is the reason why the second option is necessary: `varwidth upper=<length>`. This way, the text of the upper part of the box is put into a `varwidth` environment with a maximal width of `<length>`. You need to load the `varwidth` package [1] to use it.

This option also does not permit having a breakable `tcolorbox` or a lower part in the text. The package manual warns that it is only sensible for a `\tcbbox`, but it also works in a `tcolorbox` environment with `hbox`.

For further details, see Sections 4.17 *Capture Mode* and 4.11 *Box Content Additions* of the package manual [6].



Pay attention that `varwidth upper` sets the box width to the text width. Thus, if your title is longer than your text, you have to use some tricks.

In Box 5, to set Edna's message text to the width of the box title I used a custom option, `mywidth`, that has the title as a parameter. I took advantage of the possibility of adding some arbitrary code into an option list, using `code=<code>`.

The following passage could be a little demanding for a newbie (and also a bit boring), but if you learn these commands, you will have nice tools for typesetting your documents, quack!

I defined a new length register, using

```
\newlength{\lengthname}
```

In a register you can store a value and use it for calculations. Since I used it to store the title width, I named it `\mytitlelen`.

I also defined `\mymaxlen` as the length for the maximum width my frame can have, and with `\setlength{\lengthname}{<length>}` I set this to the `<length>` of `.9\linewidth`.

The command

```
\settowidth{\lengthname}{<some text>}
```

allows setting your length to the width of `<some text>`. I used it, in the code of the customized option `mywidth`, to set `\mytitlelen` to the width of the title text, appropriately formatted.

Then, with

```
\addtolength{\lengthname}{<length>}
```

I added the left and right space between title text and frame. To get these values, I used the lengths where `tcolorbox` stores them: `\kvtcb@lefttitle` and `\kvtcb@righttitle`. They are not mentioned in the manual, but you can find them if you look at the package code `tcolorbox.sty`. Alert: look but do not touch. Otherwise, you can do damage, quack!

Moreover, since they are internals, you should limit their use to what is strictly necessary (if the

package author changed their names, for example, your code would no longer compile).

#### Box 5 – hbox and varwidth

```
% In your preamble:
\usepackage[many]{tcolorbox}
\usepackage{varwidth}

\newlength{\mytitlelen}
\newlength{\mymaxlen}
\setlength{\mymaxlen}{.9\linewidth}

\makeatletter
\tcbset{
commonoptions/.style={
  enhanced,
  fonttitle=\bfseries\scriptsize,
  titlerule=0pt, boxrule=0pt, boxsep=0pt,
  toptitle=6pt, bottomtitle=2pt,
  opacityframe=0,
},
mywidth/.code={%
  \settowidth{\mytitlelen}{%
    \bfseries\scriptsize #1}%
  \addtolength{\mytitlelen}{%
    \kvtcb@lefttitle}%
  \addtolength{\mytitlelen}{%
    \kvtcb@righttitle}%
  \ifdimcomp{\mytitlelen}>{\mymaxlen}{%
    \tcbset{width=\mymaxlen}}{%
    \tcbset{width=\mytitlelen}}%
},
}
\makeatother

\NewTColorBox{receivedmsg}{s o m}{
  commonoptions,
  halign=flush left,
  coltitle=red, colbacktitle=yellow!30,
  colback=yellow!30,
  title={#3},
  IfBooleanTF={#1}{%
    mywidth={#3},
  }{%
    hbox, varwidth upper=\mymaxlen,
  },
  IfValueT={#2}{#2}
}

\NewTColorBox{sentmsg}{s o m}{
  commonoptions,
  halign title=flush right,
  halign=flush right,
```

```
flush right,
coltitle=blue, colbacktitle=green!30,
colback=green!30,
title={#3},
IfBooleanTF={#1}{%
  mywidth={#3},
}{%
  hbox, varwidth upper=\mymaxlen,
},
IfValueT={#2}{#2},
}

% In your document:
\begin{receivedmsg}{Tostin:}
  You know, I spent a lot of years
  disliking women. But I don't dislike
  you.
\end{receivedmsg}
\begin{sentmsg}*[show bounding box]{Edna:}
  Oh?
\end{sentmsg}
\begin{receivedmsg}{Tostin:}
  You're not a woman. You're more than a
  woman. You're a \emph{mechanic}.
\end{receivedmsg}
```

**Tostin:**

You know, I spent a lot of years disliking women. But I don't dislike you.

**Edna:**

Oh?

**Tostin:**

You're not a woman. You're more than a woman. You're a *mechanic*.

Eventually, I used `\ifdimcomp{<dimension expression>}{<relation>}{<dimension expression>}{<>true>}{<>false>}` from the `etoolbox` package [3] (`tcolorbox` loads it) to compare the width of the title to the maximum width, and, if it is greater, to limit the box width to `\mymaxlen`. I did it using the `tcolorbox` option `width=<length>`, which sets the total width of the box to `<length>`.

Please note that `\kvtcb@lefttitle` and `\kvtcb@righttitle` contain the symbol `@`. Thus, the code that uses those names must be included between `\makeatletter` and `\makeatother`. This way, L<sup>A</sup>T<sub>E</sub>X will recognize `@` as a letter when looking for the names of commands. Remember to put



`\makeatother` as soon as you no longer need this behavior. Otherwise, you can cause errors, quack!

Since I have to use `mywidth` only when the title is wider than the text, I took advantage of the possibility of testing the starred parameter with `\IfBooleanTF{⟨argument⟩}{⟨true⟩}{⟨false⟩}` and use `mywidth` when the starred `⟨argument⟩` is present (`⟨true⟩`), or `hbox` and `varwidth upper` otherwise (`⟨false⟩`).

At this point, I've set the dimension, but I also have to indicate the position. The received messages are already located to the left, the default. The sent messages should be on the right, I achieved this with the option `flush right`. Please note that this aligns *the box*, whereas `halign=flush left` aligns *the content* of the box.

The starred parameter seen above is the first #1 of the customized `tcolorbox` environments in Boxes 5 and 6. The title text is the third #3, that I set as mandatory. The second #2 is optional and I reserved it for any other setting you like to add.

With the command:

```
\IfValueT{⟨argument⟩}{⟨true⟩}
```

you can test if the optional parameter has a value and, if it does, do what is indicated in `⟨true⟩`.

I added an optional parameter to the environment `sentmsg` of Box 5: `show bounding box`.

It displays the bounding box border around your `tcolorbox`. In the example, you can see that the bounding box has the width of the line, whereas the text box is narrower. In a testing phase, showing the bounding box could be useful. Of course, it should not appear in the final document.

## 2.4 The finishing touch

The last thing to do is to add a “pointer”, to make the box appear as a callout, and the writer’s avatar.

You can easily do both things using `overlay=⟨graphical code⟩`.

This allows adding some `⟨graphical code⟩` to the box drawing process. See Section 4.12 *Overlays* of the package manual [6] for details.

I added two TikZ commands, a `\pic` with a TikZlings for the avatar and a filled `\path` for the pointer.

For the sake of brevity, I will not explain these macros here; for information, you can refer to the documentation of TikZ and TikZlings packages [5, 7].

What I would highlight is the possibility of using the `frame` coordinates. The `frame` is treated as a TikZ node. Therefore, you can use `frame.north`, `frame.north west`, and so on, to draw your graphical code.



Let me mention here the option `enlarge left by=⟨length⟩`, which enlarges the bounding box distance to the left side of the box by `⟨length⟩`. Other options for enlarging the right, top, and bottom sides of the box exist as well. See Section 4.15 *Bounding Box* of the manual [6].

In the customized style `myenlarge` of Box 6, the `⟨length⟩` is computed as the difference between `\linewidth` and the sum of the widths of the box `\kvtcb@width` and the avatar `\iconwidth`.

In this case, using `flush right` is not enough, because the avatar is not considered part of the box and, with `flush right`, it would be pushed out of the line border.

### Box 6 – Final result

```
\documentclass[twocolumn]{article}
\usepackage[many]{tcolorbox}
\usetikzlibrary{tikzlings}
\usepackage{varwidth}

\newlength{\mytitlelen}
\newlength{\mymaxlen}
\setlength{\mymaxlen}{.9\linewidth}
\newlength{\iconwidth}
\setlength{\iconwidth}{9mm}

\makeatletter
\tcbset{
  commonoptions/.style={
    enhanced,
    fonttitle=\bfseries\scriptsize,
    titlerule=0pt, boxrule=0pt, boxsep=0pt,
    toptitle=6pt, bottomtitle=2pt,
    opacityframe=0,
  },
  myenlarge/.style={
    enlarge #1 by=\linewidth -
      (\kvtcb@width + \iconwidth),
  },
  mywidth/.code={%
    \settowidth{\mytitlelen}{%
      \bfseries\scriptsize #1}%
    \addtolength{\mytitlelen}{%
      \kvtcb@lefttitle}%
    \addtolength{\mytitlelen}{%
      \kvtcb@righttitle}%
    \ifdimcomp{\mytitlelen}{>}{\mymaxlen}{%
      \tcbset{width=\mymaxlen}}{%
      \tcbset{width=\mytitlelen}}%
  },
}
\makeatother
```

```

\NewTColorBox{receivedmsg}{s o m}{
  commonoptions,
  left skip=\iconwidth,
  overlay={
    \pic[scale=.3] at
      ([shift={(-6mm,-6mm)}]]frame.north west)
    {bear};
    \path[fill=yellow!30] (frame.north) --
      ([xshift=-3mm]frame.north west) --
      ([yshift=-3mm]frame.north west) --
      cycle;
  },
  coltitle=red, colbacktitle=yellow!30,
  colback=yellow!30,
  title={#3},
  IfBooleanTF={#1}{%
    mywidth={#3},
  }{%
    hbox, varwidth upper=.8\linewidth,
  },
  align title=flush left,
  align=flush left,
  IfValueT={#2}{#2}
}

\NewTColorBox{sentmsg}{s o m}{
  commonoptions,
  right skip=\iconwidth,
  overlay={
    \pic[scale=.3] at
      ([shift={(6mm,-6mm)}]]frame.north east)
    {elephant};
    \path[fill=green!30] (frame.north) --
      ([xshift=3mm]frame.north east) --
      ([yshift=-3mm]frame.north east) --
      cycle;
  },
  coltitle=blue, colbacktitle=green!30,
  colback=green!30,
  title={#3},
  IfBooleanTF={#1}{%
    mywidth={#3},
  }{%
    hbox, varwidth upper=.8\linewidth,
  },
  myenlarge=left,
  align title=flush right,
  align=flush right,
  IfValueT={#2}{#2},
}

```

```

\begin{document}
  \begin{receivedmsg}{Jerry}
    I'm a man!
  \end{receivedmsg}
  \begin{sentmsg}{Osgood}
    Well, nobody's perfect!
  \end{sentmsg}
\end{document}

```



**Jerry**

I'm a man!

**Osgood**

Well, nobody's perfect!



### 3 Conclusions

I hope you had fun with `tcolorboxes` and remember, if you have a bug in your code:

*Chat with your duck!*

### References

- [1] D. Arseneau. The `varwidth` package, version 0.92. [ctan.org/pkg/varwidth](https://ctan.org/pkg/varwidth)
- [2] L<sup>A</sup>T<sub>E</sub>X Project Team. L<sup>A</sup>T<sub>E</sub>X for authors — current version. [ctan.org/pkg/usrguide](https://ctan.org/pkg/usrguide)
- [3] P. Lehman, J. Wright. The `etoolbox` Package: An  $\epsilon$ -T<sub>E</sub>X Toolbox for Class and Package Authors, version v2.5k (2020-10-05). [ctan.org/pkg/etoolbox](https://ctan.org/pkg/etoolbox)
- [4] C. Maggi. The DuckBoat—Beginners' Pond: No more table nightmares with `tabularray`! *TUGboat*, 44(1):64–70, 2023. [tug.org/TUGboat/tb44-1/tb136duck-tabularray.pdf](https://tug.org/TUGboat/tb44-1/tb136duck-tabularray.pdf)
- [5] samcarter. The `TikZlings` package — drawing animals and beings in `TikZ`, version 1.0 (2022-06-17). [ctan.org/pkg/tikzlings](https://ctan.org/pkg/tikzlings)
- [6] T.F. Sturm. The `tcolorbox` package, version 6.2.0 (2024-01-10). [ctan.org/pkg/tcolorbox](https://ctan.org/pkg/tcolorbox)
- [7] T. Tantau, et al. The `TikZ` and `PGF` Packages. [ctan.org/pkg/pgf](https://ctan.org/pkg/pgf)

◇ Herr Professor Paulinho van Duck  
 Quack University Campus  
 Sempione Park Pond  
 Milano, Italy  
 paulinho dot vanduck (at) gmail  
 dot com