The PracTEX Journal, 2010, No. 2
Article revision 2010/10/23

Scientific Presentations with LATEX

Authors Marius Hofert[1], Markus Kohm[2]

Abstract In this article, we show how scientific presentations can be created based on the KOMA-Script document class `scrartcl`. The main advantage of the suggested approach is that the presentation slides allow for easy copy-and-paste of content from other LATEX documents such as research papers or handouts. Therefore, presentations slides are quickly generated, without the author having to learn how to use other LATEX presentation packages. Additionally, in contrast to the rather overused templates of the more common presentation packages, the slides can be individually created and thus tailored to the topic of the presentation.

1 Introduction

Are you a scientific researcher or lecturer working in the field of mathematics or in one of the natural sciences? Is most of your time consumed by lecturing, researching, and presenting your ideas and results at conferences? Are you using LATEX to create your scientific documents such as research papers and presentation slides? (If you are not using LATEX for this, please stop reading this article and consult a LATEX tutorial to learn why you should.) If you answered 'Yes' to any of these questions, then we recommend that you spend a couple of minutes reading this article. It will save you a lot of time in the future.

This article shows how scientific presentations can be

- created quickly without the need to learn the complicated commands of specialized LATEX packages, and
- individually designed, so that you will have the 'wow factor' on your side when giving them.

[1] RiskLab, Department of Mathematics, ETH Zurich, 8092 Zurich, Switzerland, marius.hofert@math.ethz.ch. The author (Willis Research Fellow) thanks Willis Re for financial support while this work was being completed.

[2] KOMA-Script Documentation Project, http://www.komascript.de (primarily in German), komascript@gmx.info

Copyright © 2010 Marius Hofert, Markus Kohm.
Permission is granted to distribute verbatim copies
of this document provided this notice remains intact.

There are many LaTeX packages for creating presentation slides, including TeXPower/Powersem, Seminar, Ppower4, Prosper, FoilTeX, PDFscreen, and Beamer. As a researcher using LaTeX, you are probably familiar one or more of these (especially Beamer, which is arguably the most popular of these packages within the scientific community). We feel that the standard templates provided by these packages are overused. Indeed, we find it quite boring to see the same default template over and over again at conferences.

We also feel that some of the more elaborate features provided by these packages are overused. One good example of this is the overlay feature which reveals each of the items of a list in sequence. In our experience, the lighting conditions at conferences are often such that one either does not see that other items will follow or that one can read the full list right away. We have seen many talks where the interaction required for each list item caused the speakers' flow to be broken, thus making them quite difficult to follow.

There are many good books that deal with the topic of how to create great presentations. We recommend Tufte (2007), Reynolds (2008), and Duarte (2008). Although the mentioned concepts and techniques only partly apply to scientific (as opposed to, e.g., marketing) presentations, they may help you to enhance your presentations. First and foremost, you should spend time on thinking about the main idea of your presentation and how it can be best understood by the audience; otherwise, your presentation will be lost time for your audience. Then (and only then), create the slides. Afterwards, rehearse and improve your presentation over and over again. Always keep in mind that if your audience consists of thirty people, a twenty-minute talk is just twenty minutes for you, but ten hours for your audience all together! Usually, the problem is that too much time is spent on preparing the slides, so there is little time left to rehearse and improve the presentation itself.

Having spoken about how to create presentations, let us consider the typical workflow of a researcher. Before going to a scientific conference you will have probably already prepared a working paper or an article on the topic you are going to talk about. In many cases, you will have already submitted an article to a scientific journal for publication. On other occasions, you might even write an article shortly after a conference, e.g., for a conference proceeding. Regardless, to save time, you want to copy-and-paste the main formulas and results you have typed in LaTeX from one document to another (just having to adjust minor details).

Of course, this can be done with the aforementioned packages. Usually, however, many commands have to be inserted, replaced, or removed, e.g., commands for the beginning and the end of a new slide or special commands for fancy lists.

2 The idea

Our idea is to create presentations using a standard document class rather than special presentation packages and classes. Since we mostly work with KOMA-Script, we decided to use the `scrartcl` document class for our presentation slides. The most important thing to note is that, in principle, you can copy-and-paste everything (text, lists, tables, figures, video clips, etc.) you have prepared in other LaTeX documents, such as research papers or handouts, to your presentation file and then you can make any necessary changes to the content.

Listing 1 shows the document class and options for our presentation slides. The paper and font size are chosen as with the `Beamer` document class. Note that these sizes can be adjusted for various projector capabilities, including 144 mm : 90 mm (16 : 10), 120 mm : 96 mm (16 : 9), 160 mm : 90 mm (HDTV 720 p/i), and 192 mm : 108 mm (HDTV 1080 p/i).

```latex
\documentclass[
   paper=128mm:96mm,% like beamer
   fontsize=11pt,% like beamer
   pagesize,% write page size to dvi or pdf
   parskip=half-,% paragraphs are separated by half a line
   numbers=noendperiod,% no periods after section numbers
   captions=nooneline% same treatment of one/several lines
]{scrartcl}
\linespread{1.12}% enlarge line space
```

Listing 1 Document class, options, and adjusted line spread.

In order to improve the readability of the slides, we enlarge the line space by using `\linespread` (see Listing 1). Next, we load several packages: `fontenc` (with option `T1`), `lmodern`, `babel` (e.g., with option `american`), `microtype`, and `xcolor`. We then load the `geometry` package with suitable options to set up the page — margins, type area, etc. (see Listing 2). In principal, the presentation slides can now be viewed as a standard scrartcl document which we later fill with content, formulas, figures, and so on. The packages scrpage2 and titlesec will be used later for adjusting other aspects of the page style and the distance between headings and text, respectively. The package tocstyle will be used for adjusting the table of contents, which will give an outline of the presentation.

```latex
\usepackage{calc}% working with lengths, counters etc.
\usepackage[includeheadfoot,top=3.5mm,bottom=3.5mm,left=5.5mm,right=5.5mm,headsep=6.5mm,footskip=8.5mm]{geometry}% set page layout parameters
\usepackage{scrpage2}% package for page style with not only uppercase letters in the head
\usepackage{titlesec}% for reducing space between ((sub)sub)sections and text
\usepackage{tocstyle}% for adjusting table of contents
```

Listing 2 Packages and options for adjusting the page structure.

Next, we load our favorite packages, e.g., amsthm for nice theorem environments, bm for bold math symbols, enumitem for creating nice lists, graphicx for including graphics, tikz for creating sophisticated drawings, tabularx and booktabs for creating nice tables, biblatex for creating sophisticated bibliographies, and hyperref for creating links. The option pdfpagemode=FullScreen can be used for the hyperref package so that Adobe Reader will open the document in full screen mode — quite a handy feature for giving talks.

Now, let us turn to the settings for the page style, which are given in Listing 3.

```latex
% page style
\pagestyle{scrheadings}% activates pagestyle from scrpage2
\clearscrheadfoot% clear head and foot
\setkomafont{pageheadfoot}{\normalfont\color{white}\sffamily}% setting for page head and foot
% optical vertical centering of page contents
\makeatletter
\renewcommand*{\@textbottom}{\vskip \z@ \@plus 1fil}
\newcommand*{\@texttop}{\vskip \z@ \@plus .5fil}
\addtolength{\parskip}{\z@\@plus .25fil}% stretch parskip
```

```
\makeatother
```

Listing 3 Settings for the page style and for vertically centering the page content.

Looking at the commands above, we first activate the page style from `scrpage2` using the `\pagestyle{scrheadings}` command and clear the page head and foot via `\clearscrheadfoot`. Later on, we will add bars to the top and bottom of the each page. Since these bars are intended to separate information about the talk and the presenter from the actual content of the talk, we will choose a dark background color and set the text colour to white for these bars. We therefore need to adjust settings for the page head and foot accordingly using the command `\setkomafont{pageheadfoot}{...}`. Listing 3 also shows how the page content can be centered vertically.

Next, we adjust the spacings between ((sub)sub)sections and the text. In particular, we use `\titlespacing{...}{0mm}{0mm}{x mm}`, where `x` is chosen as 0 mm, -1 mm, and -2 mm and the dots are `\section`, `\subsection`, and `\subsubsection`, respectively.

Now we come to the most important part. (Note that what we demonstrate here is just one of many possible examples.) As mentioned before, we would like to have, as default, horizontal bars at the head and foot of each slide. The top bar should show the title — which can easily be replaced, e.g., by a running section heading — and the bottom bar the name of the authors, their affiliation, and the page number. (A page number makes sense for the audience to make comments and ask questions later.) So far so good; but how can we build these bars? The solution is to use the power of `tikz` in combination with the easy-to-use page-style definitions provided by `scrpage2`. Listing 4 gives the details for the page head.

```
\ihead{% head left
   \hspace{-2mm}%
   \begin{tikzpicture}[remember picture,overlay]
      \node [xshift=\paperwidth/2,yshift=-\headheight] (
         mybar) at (current page.north west)[rectangle,
         fill,inner sep=0pt,minimum width=\paperwidth,
         minimum height=2\headheight,top color=mybgcolor
         !64,bottom color=mybgcolor]{};% bar
      \node[below of=mybar,yshift=3.3mm,rectangle,shade,
        inner sep=0pt,minimum width=128mm,minimum height
        =1.5mm,top color=black!50,bottom color=white]{};%
         shadow
    \end{tikzpicture}%
    \myhead
}
```

Listing 4 Definition of the page head.

We use a `tikzpicture` environment with options `remember picture` and `overlay` to draw the bar. The upper left corner of the page serves as a reference point for the center of the rectangle. We then just shift the rectangle with `xshift` and `yshift` so that it is drawn where we want it. We use a vertical gradient for the rectangle based on the background color `mybgcolor`, which is a nice color we defined in the document preamble. The second `node` is a drop shadow for the top bar. Note that we shift the head two millimeters towards the margin.

The bar at the bottom of the page is realized similarly, see Listing 5.

```
\newlength{\footheight}
\setlength{\footheight}{8mm}
\addtokomafont{pagefoot}{\footnotesize}% size for foot
\setkomafont{pagenumber}{\color{white}}% white page number
\ifoot{% foot left
    \hspace{-2mm}%
    \begin{tikzpicture}[remember picture,overlay]
        \node [xshift=\paperwidth/2,yshift=\footheight/2] at
            (current page.south west)[rectangle,fill,inner
           sep=0pt,minimum width=\paperwidth,minimum height
           =\footheight,top color=mybgcolor!64,bottom color=
           mybgcolor]{};% bar
    \end{tikzpicture}%
    \myauthor\ \raisebox{0.2mm}{$\bm{\vert}$}\ \myuni
}
\ofoot[\pagemark\hspace{-2mm}]{\pagemark\hspace{-2mm}}%
   foot right (plain pages do also have page numbers)
```

Listing 5 Definition of the page foot.

Here, we use the lower left corner as the reference point for the bar and move the author information as well as the page number two millimeters to the respective margins, similar to the adjustments for the page head.

Remark
Note that you have the full power and freedom provided by tikz to create bars in any way you like: they can be specifically designed for a certain topic of the talk, they can contain a logo, horizontal or vertical gradients, or even pictures you have taken with your camera. The tikz package has an excellent manual that provides many examples of how to create nice drawings. A bit of experimentation will generally lead to nice headers and footers for presentation slides. Further, you can be almost sure that none of your colleagues uses the same design.

Next, let us consider the table of contents. Since the slides are based on the article class, we can have LaTeX create a table of contents for us that gives an outline of the presentation. The name of the table of contents can be changed with the command \AtBeginDocument{\renewcaptionname{american}{\contentsname}{...}}, where the dots stand for the your chosen name. We would also like to remove the dots connecting the ((sub)sub)sections with their page numbers. For this we use the previously loaded package tocstyle to define a new style for the table of contents (see Listing 6).

```
\AtBeginDocument{\renewcaptionname{american}{\contentsname}{\large Outline}}% change name of toc
\makeatletter
\newtocstyle[noonewithdot]{nodotnopagenumber}{% define tocstyle without dots and page numbers
  \settocfeature{pagenumberbox}{\@gobble}%
}
\makeatother
\usetocstyle{nodotnopagenumber}
```

Listing 6 Settings for the table of contents.

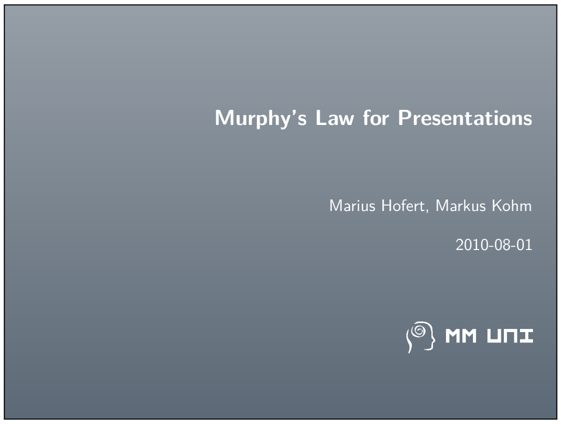
That's pretty much it! Oh wait, we forgot to reveal how the first and last slides can be designed. Well, this is simple: Use \thispagestyle{plain} to remove the running bars or \thispagestyle{empty} to remove the page numbers and the running bars. You are now free to use a tikzpicture environment (with options remember picture and overlay as before) to design the pages as you want.

Remark
The package tikz can also be used to design boxes which you can use to wrap environments such as important theorems or remarks. Further, you can even design your own arrows and bullet points using tikz. This, together with the enumerate environment, enables you to easily customize your presentation's lists. We implemented both of these features in our example presentation in the following section. There are many benefits to our approach to creating presentation slides: You can freely and individually design the slides this way without being restricted to fixed commands given in the aforementioned LaTeX presentation packages. You are not required to learn any new commands, e.g., commands to begin or end a slide. You can fill the slides with content in the same way as you would write an article (if the page is full, the next is used) and you can copy-and-paste text, figures, etc. from other LaTeX documents to your presentation document. If you want to continue to the next slide, simply use the command \clearpage.

One final note: the approach described in the last section gives the slide designer a lot of freedom, but this should be exercised with care. It is clear that, for example, one should not use light yellow text color on a white background. Please refer to the aforementioned references for avoiding mistakes like this.

3 An example presentation

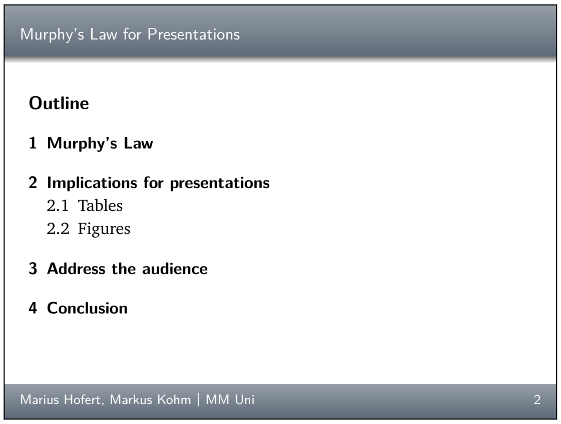
The example below, which shows the approach described in the last section for designing presentation slides, is intentionally simple. For the typeface, we chose Charter (via the package charter). Even though this typeface has serifs it is suitable for both printing and displaying in low resolution. In order to have colors that fit together nicely, we recommend using a tool for creating color palettes, e.g., http://kuler.adobe.com/#create/fromacolor. There are many ways to customize your slides. For example, the package ocgtools allows to insert optional content groups in PDFs, the package listings may be used to include source code, and the package Sweave allows to execute source code written in the statistical software R and display the output on the presentation slides right away. The package movie15 even allows the inclusion of video clips.



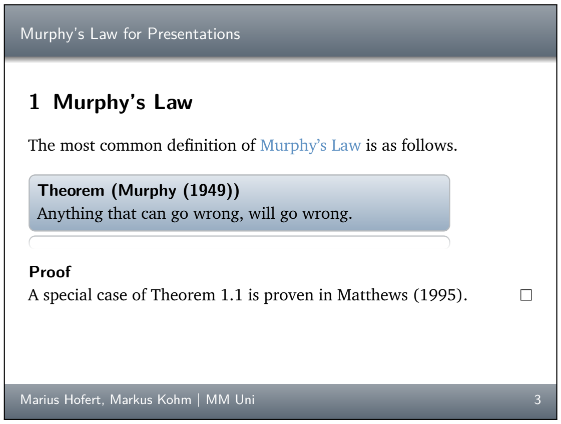
Murphy's Law for Presentations

Marius Hofert, Markus Kohm

2010-08-01



Outline

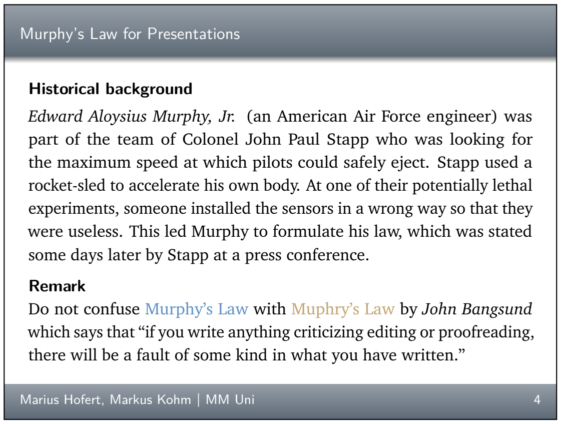


1 Murphy's Law

The most common definition of Murphy's Law is as follows.

Theorem (Murphy (1949))
Anything that can go wrong, will go wrong.

Proof
A special case of Theorem 1.1 is proven in Matthews (1995). □

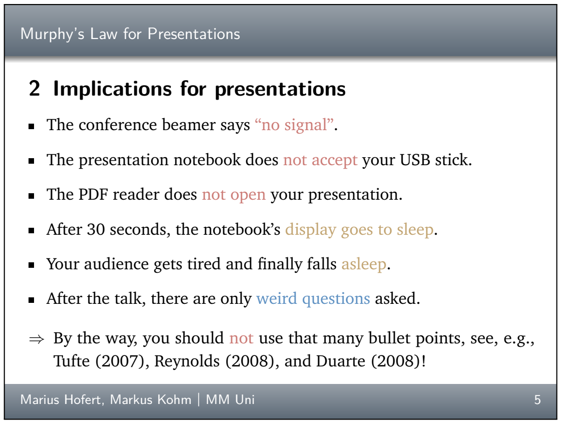


Historical background

Edward Aloysius Murphy, Jr. (an American Air Force engineer) was part of the team of Colonel John Paul Stapp who was looking for the maximum speed at which pilots could safely eject. Stapp used a rocket-sled to accelerate his own body. At one of their potentially lethal experiments, someone installed the sensors in a wrong way so that they were useless. This led Murphy to formulate his law, which was stated some days later by Stapp at a press conference.

Remark

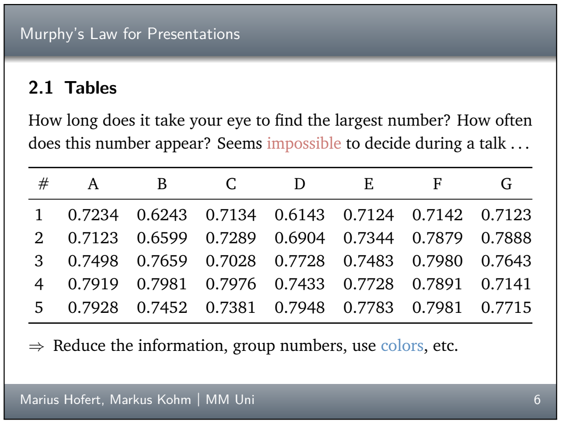
Do not confuse Murphy's Law with Muphry's Law by *John Bangsund* which says that "if you write anything criticizing editing or proofreading, there will be a fault of some kind in what you have written."



2 Implications for presentations

- The conference beamer says "no signal".
- The presentation notebook does not accept your USB stick.
- The PDF reader does not open your presentation.
- After 30 seconds, the notebook's display goes to sleep.
- Your audience gets tired and finally falls asleep.
- After the talk, there are only weird questions asked.

⇒ By the way, you should not use that many bullet points, see, e.g., Tufte (2007), Reynolds (2008), and Duarte (2008)!

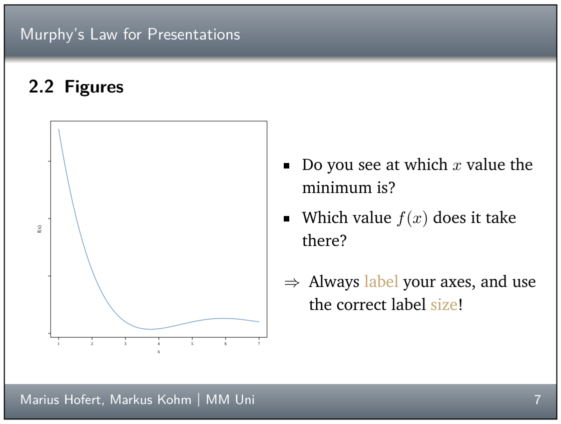


2.1 Tables

How long does it take your eye to find the largest number? How often does this number appear? Seems impossible to decide during a talk . . .

#	A	B	C	D	E	F	G
1	0.7234	0.6243	0.7134	0.6143	0.7124	0.7142	0.7123
2	0.7123	0.6599	0.7289	0.6904	0.7344	0.7879	0.7888
3	0.7498	0.7659	0.7028	0.7728	0.7483	0.7980	0.7643
4	0.7919	0.7981	0.7976	0.7433	0.7728	0.7891	0.7141
5	0.7928	0.7452	0.7381	0.7948	0.7783	0.7981	0.7715

⇒ Reduce the information, group numbers, use colors, etc.



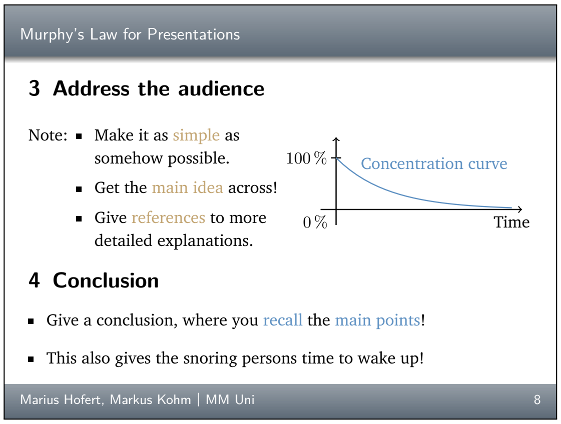
Murphy's Law for Presentations

2.2 Figures

- Do you see at which x value the minimum is?
- Which value $f(x)$ does it take there?

⇒ Always label your axes, and use the correct label size!

Marius Hofert, Markus Kohm | MM Uni 7



Murphy's Law for Presentations

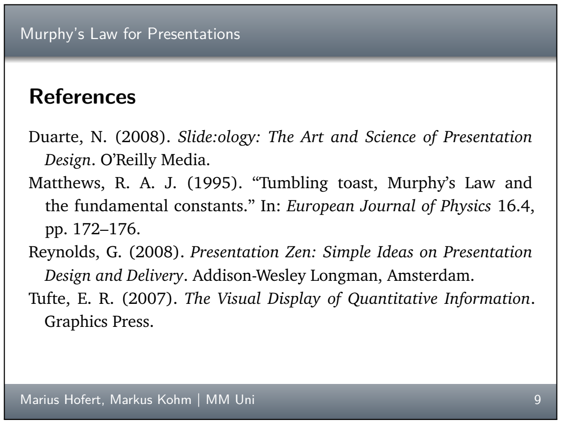
3 Address the audience

Note:
- Make it as simple as somehow possible.
- Get the main idea across!
- Give references to more detailed explanations.

4 Conclusion

- Give a conclusion, where you recall the main points!
- This also gives the snoring persons time to wake up!

Marius Hofert, Markus Kohm | MM Uni 8



Murphy's Law for Presentations

References

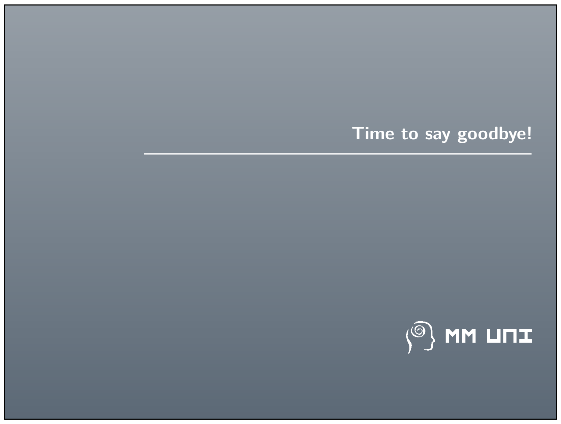
Duarte, N. (2008). *Slide:ology: The Art and Science of Presentation Design*. O'Reilly Media.

Matthews, R. A. J. (1995). "Tumbling toast, Murphy's Law and the fundamental constants." In: *European Journal of Physics* 16.4, pp. 172–176.

Reynolds, G. (2008). *Presentation Zen: Simple Ideas on Presentation Design and Delivery*. Addison-Wesley Longman, Amsterdam.

Tufte, E. R. (2007). *The Visual Display of Quantitative Information*. Graphics Press.

Marius Hofert, Markus Kohm | MM Uni 9



Time to say goodbye!

MM UNI

4 Conclusion

In this article, we showed one way in which presentation slides based on the KOMA-Script document class `scrartcl` can be created. The presentation document allows the author to easily copy-and-paste from other LaTeX documents, such as research papers, and can therefore be generated quite quickly. The flexibility of the individual slide design can lead to interesting and personalized presentations. The same general principle can be applied to creating scientific posters. We hope that the scientific community realizes these advantages and its members start to create their LaTeX documents with more individual style, so that we do not have to see the same templates over and over again.

References

Duarte, N. (2008), “Slide:ology: The Art and Science of Presentation Design”, O’Reilly Media.

Reynolds, G. (2008), “Presentation Zen: Simple Ideas on Presentation Design and Delivery”, Addison-Wesley Longman, Amsterdam.

Tufte, E. R. (2007), “The Visual Display of Quantitative Information”, Graphics Press.