

# How To Do A Talk In T<sub>E</sub>X

## One Of Many Solutions

Version 2.0

Matthias Meister

Regensburg, October 2010

- Make it possible to prepare presentations in T<sub>E</sub>X
- Without having to learn lots of extra syntax
- By using simple macros that can easily be adapted to one's needs, maybe for each presentation prepared
- Without restricting the possibilities that T<sub>E</sub>X offers

You need the program `pdftex`, which probably is included in your T<sub>E</sub>X-distribution.

You need the file `present.tex`, which is to be `\input` at the beginning of your source code.

You furthermore need a pdf-viewer with fullscreen display capabilities, e.g. `xpdf`.

**Warning:** Some pdf-viewers do not handle links in a document properly.

Version 2.0 does no longer require the files `miniltx.tex`, `color.tex`, `color.sty`.

You can type text as usual, inline equations  $a^2 + b^2 = c^2$ ,  
displayed equations

$$\exp(z) = \sum_{n=0}^{\infty} \frac{z^n}{n!} = \lim_{n \rightarrow \infty} \left(1 + \frac{z}{n}\right)^n$$

and tables

	$x$	$\Delta x$
A	1.03	0.07
B	2.05	0.06

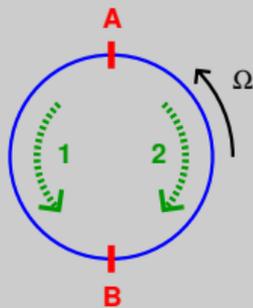
So you probably can use  $\text{T}_{\text{E}}\text{X}$  in the ordinary fashion; just run [pdftex](#) on your source instead of [tex](#).

You can include images easily:



A pixel image (png)

---



A vector image (pdf)

---

You type text, equations, and so on as usual in  $\text{T}\text{E}\text{X}$ . Keep in mind, though, that the effective paper size is rather small (12cm wide, 9cm high in standard configuration).

The paper size is small, because you rely on the fullscreen mode of your pdf-viewer to blow the slide up to full screen size, thus also automatically enlarging the fonts.

Your macros should work (unless there is a collision of names), because what you are doing is creating an ordinary pdf-file with [pdftex](#), for viewing it with a pdf-viewer.

`present.tex` defines the following fonts:

Font Command	Purpose	Default Value
<code>\titlefont</code>	presentation title	<code>cmssbx10</code> at 20pt
<code>\slidetitlefont</code>	title of a slide	<code>cmssbx10</code>
<code>\normalfont</code>	ordinary text	<code>cmss12</code>
<code>\linkbarfont</code>	text in <code>\LinkBar</code>	<code>cmss8</code>
<code>\it</code>	italic text	<code>cmti12</code>
<code>\rm</code>	roman text in math	<code>cmr</code>

Of course you can define further font commands or redefine the existing ones. The default fonts are used in this presentation, so you see what they look like.

Colours are defined as control sequences, e.g.

```
\def\mycolour{R G B}
```

where *R G B* are the red, green, and blue values of the colour defined, with  $0 \leq R, G, B \leq 1$ .

Colours are used in these ways:

```
\setcolour\mycolour
```

to set the colour of following text , grouping is respected

```
\coloured\mycolour{Stuff}
```

to set *Stuff* in colour `\mycolour`

```
\setbgcolour\mycolour
```

to set the colour of the background

The following colours are defined in `present.tex`:

`\backgroundcolour`, for the slide background

`\textcolour`, for the text, and

`\attentioncolour`, an **attention** colour.

You draw attention to **Stuff** by saying `\att{Stuff}`.

If a predefined colour is redefined, `\setcolour` must also be used for the change to take effect for text.

Colour handling is different from version 1.0 of `present.tex`. The additional files `miniltx.tex`, `color.tex`, `color.sty` are **no longer** required.

The paper size is determined by the dimensions `\pdfpagewidth` and `\pdfpageheight`.

If you say `\StandardAspect`, you get a page 120mm wide and 90mm high.

If you say `\WideAspect`, you get a page 144mm wide and 90mm high.



Images can be included with

```
\image[dimensions]{filename}
```

where *dimensions* are **height**, **depth**, and **width**, familiar from T<sub>E</sub>X.



If only **width** is given, the image is scaled with the aspect ratio preserved.

Image files need to be in a format that can be handled by the pdf-viewer and by **pdftex**. E.g. pdf, png, jpg should work.

Each slide has a headline, a body, and a footline.

The headline holds the slide title, which is set by

`\SlideTitle{TITLE}`

The footline is defined by `\SlideFoot`, of which several versions are contained in `present.tex`. Uncomment the one you want, or define further ones.

The version used here shows the number of the current slide and the total number of slides in the centre (the latter is provided by the `\LP` macro), and the `\PageBar` on the right. The `\PageBar`-symbols `◀ ▶ ↻ ↺` move to the previous or following page, or move back and forth in the page history.

The page history is relevant if cross-references are used in the presentation. These will be discussed subsequently.

The page history is relevant if cross-references are used in the presentation. These will be discussed subsequently.

With cross-references or links, also the `\LinkBar` can be useful. In the `\SlideFoot`-version used, it is shown on the left side of the footline, but currently is defined to be empty.

The page history is relevant if cross-references are used in the presentation. These will be discussed subsequently.

With cross-references or links, also the `\LinkBar` can be useful. In the `\SlideFoot`-version used, it is shown on the left side of the footline, but currently is defined to be empty.

The `\LinkBar` is intended to hold links to various places in the presentation you might want to jump to.

The page history is relevant if cross-references are used in the presentation. These will be discussed subsequently.

With cross-references or links, also the `\LinkBar` can be useful. In the `\SlideFoot`-version used, it is shown on the left side of the footline, but currently is defined to be empty.

The `\LinkBar` is intended to hold links to various places in the presentation you might want to jump to.

Before we turn to cross-references or links: You start a new slide by saying `\NewSlide`. If instead you say `\NewFrame`, it has almost the same effect, only the slide number doesn't get increased. This is useful, if a slide is to be shown incrementally.

---

`\target{NAME}` creates a target named *NAME* for a link at the position in the presentation where it is used.

`\link{NAME}{Stuff}` turns *Stuff* into a link to the target named *NAME*.

`\weblink{URI}{Stuff}` makes *Stuff* a link to the specified *URI*.

`\filelink{file}{filedest}{Stuff}` makes *Stuff* a link to destination *filedest* in *file*.

*filedest* for example can be `[page /Fit]`, with *page* the page number (starting at 0).

A further possibility can be found here (click).

```
\linkarea{dest}{rect}{border}{colour}
```

Creates a rectangular area which is a link to target *dest*; *rect* consists of four space-separated numbers for lower left and upper right corner, *border* is the border width, and *colour* is the border colour, specified as three space-separated values for red, green, and blue, all between 0 and 1.



Notice also that `\LinkBar` has been redefined (on the previous slide already)

You can set images for the background with

```
\SetBackground{filename}
```

where *filename* is the name of the image file.

The background image is unset (i.e. the background colour will show again) with `\UnsetBackground`.

The background image is scaled to width and height of the slide. It should have an appropriate aspect ratio.

As you see some combinations of text colour and background image can cause visibility problems.

PDF provides the possibility to include movies, and to launch applications (which could be a movie player) via clickable elements. The usability for a presentation depends on how to manage the player together with a fullscreen display of the slides. Movie support is not officially included in `present.tex`.

If you prepare a pdf-image for a presentation which is to be shown on some different computer, it would be best to embed fonts used in the image into the image file. For example, if you have a file `image.eps`, you need to convert it to pdf, in order to use it with `pdftex`. You can embed the fonts by

```
ps2pdf -dEPSCrop=true -dPDFFA image.eps
```

A look at `present.tex` is recommended.