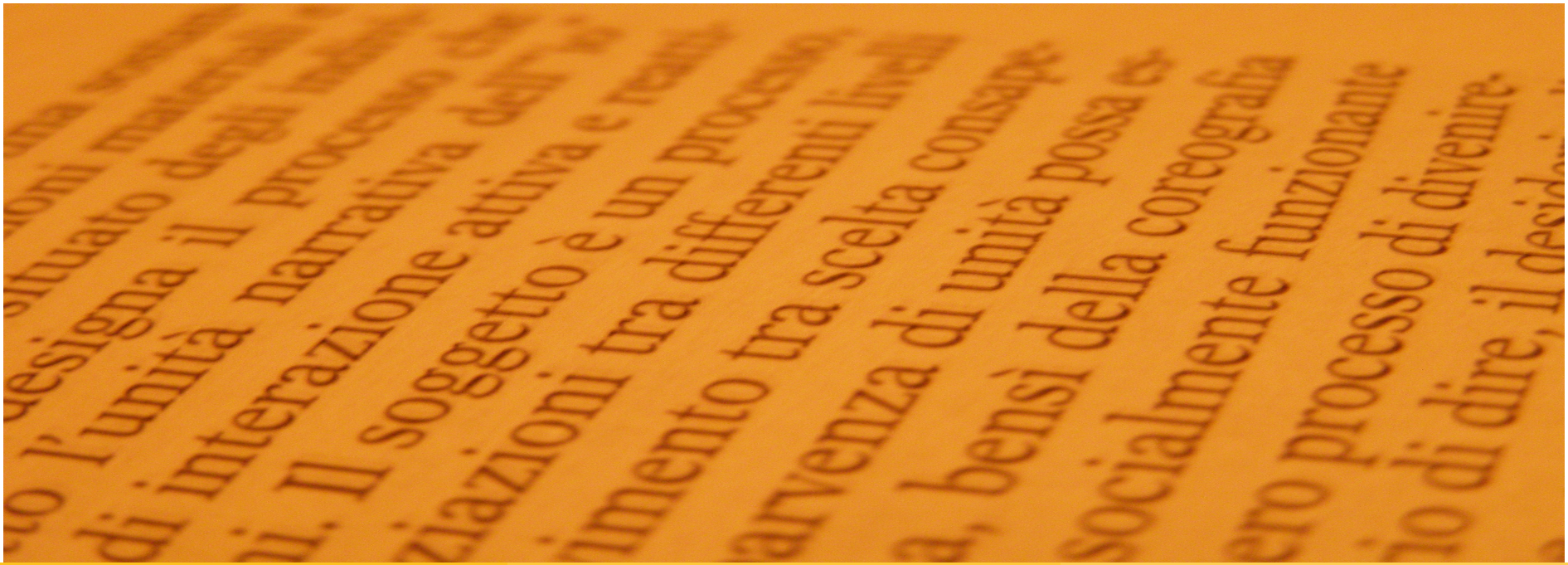
Org Beamer reference card

Fabrice Niessen

January 29, 2017



Plan

 <img
src="http://img.shields.io/:license-gpl-blue.svg"
alt=":license-gpl-blue.svg" />
<a href="https://www.paypal.com/cgi-bin/webscr?cmd=
_donations&business=VCVAS6KPDQ4JC&lc=BE&item_number=
refcard%2dorg%2dbeamer&currency_code=EUR&bn=PP%
2dDonationsBF%3abtn_donate_LG%2egif%3aNonHosted"> <img
src="https:
//www.paypalobjects.com/en_US/i/btn/btn_donate_LG.gif"
alt="btn_donate_LG.gif" />

Description

Welcome to Org Beamer reference card. It contains the reference documentation that describes how to write presentations based on:

- Org mode 8 and
- the LaTeX Beamer class.

Those free tools allow you to easily produce high quality PDF files which are going to be displayed on *every* computer exactly the way they looked on *your* computer.

Features

Since they are like any other Org mode document, authoring presentations with Org Beamer is very easy, thanks to its straightforward syntax.
The obvious advantages of this approach are that:

- you don't have to know LaTeX in order to create Beamer presentations.
- you are more productive when editing presentations:
  - you can expand and collapse slides,
  - you can switch quickly between slides by using navigation (speed) commands,
  - you can incorporate code blocks (in R or in many other languages) for illustration, and evaluate them to actually render output (including plots).

Requirements

- A working LaTeX installation is required for exporting to PDF. If it is not yet installed on your system, install TeX Live (for example).
- You must define a `beamer` class in `org-latex-export-classes`:

```
(eval-after-load "ox-latex"

  ;; update the list of LaTeX classes and associated header (encoding, etc.)
  ;; and structure
  '(add-to-list 'org-latex-classes
                `("beamer"
                  ,(concat "\\documentclass[presentation]{beamer}\n"
                           "[DEFAULT-PACKAGES]"
                           "[PACKAGES]"
                           "[EXTRA]\n")
                  ("\\section{%s}" . "\\section*{%s}")
                  ("\\subsection{%s}" . "\\subsection*{%s}")
                  ("\\subsubsection{%s}" . "\\subsubsection*{%s}"))))
```

- For nice code blocks, use Listings instead of Verbatim:

```
(setq org-latex-listings t)
```

Creating a title page I

The very first slide (called frame in a Beamer presentation) is the title page. By default, it will automatically be displayed with the following elements:

- the document title

```
#+TITLE: Document title
```

  (file name, if none specified – could be removed in the future)

- the author(s)'s name

```
#+AUTHOR: John Doe
```

  (Emacs Lisp variable `user-full-name`, if none specified)

- a date

```
#+DATE: 2017-01-01
```

Creating a title page II

(LaTeX macro `\today`, if none specified)

The insertion of `\author` can be turned off with:

```
#+OPTIONS: author:nil
```

Creating a title page III

The author's email can be included with:

```
#+AUTHOR:     \href{mailto:email@example.com}{John Doe}
#+AUTHOR:     \texorpdfstring{John Doe\newline\url{email@example.com}}{John Doe} % D
#+BEAMER_HEADER: \author{\texorpdfstring{John Doe\newline\url{email@example.com}}{J
```

Other elements:

- the document subtitle,
- their affiliation (institute), and
- a title graphic

can be included with the following commands:

```
#+BEAMER_HEADER: \subtitle{Document subtitle}
#+BEAMER_HEADER: \institute[INST]{Institute\\\url{http://www.institute.edu}}
#+BEAMER_HEADER: \titlegraphic{\includegraphics[height=1.5cm]{InstLogo}}
```

XXX Why do I have to use :eval no (in Org blocks)?
The inner theme dictates how the title page is rendered.

```
#+BEAMER_HEADER: \logo{\includegraphics[height=.9cm]{InstLogo}}
```

Structure basics

Org mode presentations contain headings at different levels. By default,

- Headings at the first outline level will become titles of the different frames.
- Deeper levels will be used as structural environments.
- The table of contents frame is created but is blank (you'll understand why later).
  You can remove it by setting the `toc` option (default: `t`) from the `#+OPTIONS:` keyword to `nil`:

```
#+OPTIONS: toc:nil

* Frame 1

Some content.

** Block

This is a block.
```

Creating a simple frame

To create a frame with bullets, you simply use standard Org mode bullets:

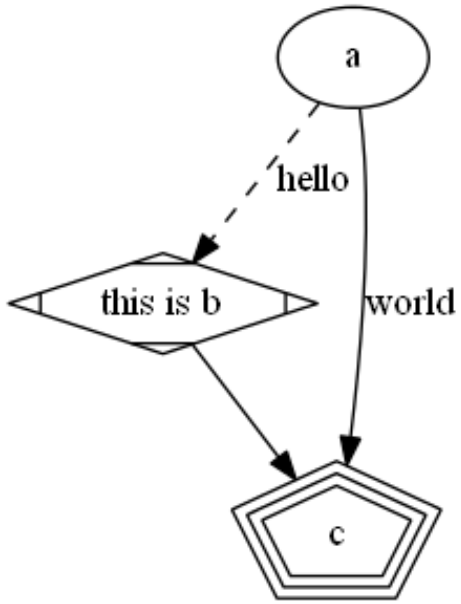
```
* A title
  #+Beamer: \framesubtitle{A subtitle}

- Bullet 1
- Bullet 2
- Bullet 3
```

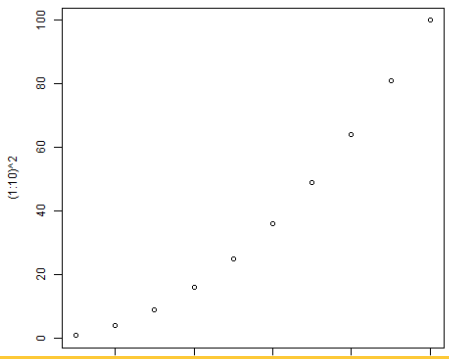
Content within frames is formatted using standard Org mode syntax.
The optional subtitle does not have an Org syntax because it's specific to the Beamer back-end only.

Using graphics

How to center pictures horizontally?

- Figures





Using graphics and text

Look at `wrapfig` or `parinc`

Frame with code

Here is a simple R code block. . .

```
#+begin_src R :exports both
summary(cars)
#+end_src
```

. . . that will display the code and show its output in the frame:

```
summary(cars)
```

XXX Is this needed?

```
#+LATEX_HEADER: \lstdefinelanguage{R}{}
```

Frame with code only

To display a code block without evaluating it, you specify the `:eval no` option:

```
#+begin_src R :eval no
summary(cars)
#+end_src
```

It only renders the code:

```
summary(cars)
```

Frame with output only

To display the output of a code block without echoing the underlying code, you specify the `:exports results` option:

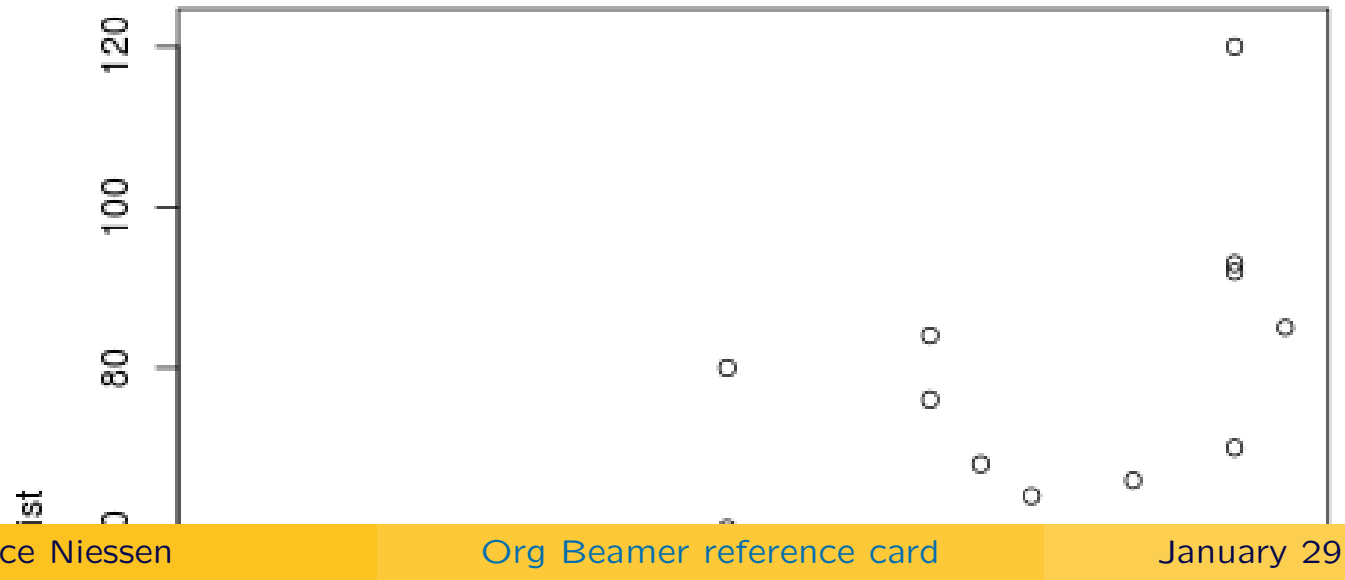
```
#+begin_src R :exports results
summary(cars)
#+end_src
```

It only renders the results:

Frame with plot

Code blocks can also be used to include plots within presentations.
To display a plot while omitting the code used to generate it, you can include a code block like this:

```
#+begin_src R :exports results :results graphics :file qplot.png
plot(cars)
#+end_src
```



Creating a table of contents I

If you set the `H` option from the `#+OPTIONS:` keyword (or the `org-beamer-frame-level` variable) to 2:

```
#+OPTIONS: H:2 toc:t
```

then:

- First-level headings become sections listed in the table of contents[1], and
- Second-level headings become the frames.

XXX Misplaced footnote!

Creating a table of contents II

If you set the `H` value to `3` in the `OPTIONS` line:

```
#+OPTIONS: H:3 toc:t
```

then:

- First- and second-level headings become sections and subsections listed in the table of contents, and
- Third-level headings become the frames.

In many themes, sections (and subsections, when `H:3`) appear in the sidebar or heading.

[1] The items in the TOC are clickable and take you to specific frames in the presentation.

Present a bibliography

Exporting to PDF

Type:

```
M-x load-library RET ox-beamer RET
```

to load the Beamer back-end library, and to obtain extra commands in the LaTeX export menu:

`C-c C-e l B` Export as LaTeX buffer (Beamer).

`C-c C-e l b` Export as LaTeX file (Beamer).

`C-c C-e l P` Export as PDF file (Beamer).

`C-c C-e l O` Export as PDF file and open (Beamer).

ou do not need to set `LaTeX_CLASS` to `beamer`.

Support editing

Type:

```
M-x org-beamer-mode RET
```

to load the minor mode `org-beamer-mode` easing the edition of the document structure (through the key binding `C-c C-b`, which offers fast selection of a Beamer environment).
You can also turn it on with:

```
#+STARTUP: beamer
```

in your document.

Label!!!

Label to theorem.

Create a handout

You can distribute your presentation in the form of handouts. Presentations exported in this manner are entirely animation-free: overlays are removed and just the last "slide" of each frame is printed.

```
#+LATEX_CLASS_OPTIONS: [handout]

#+LATEX_HEADER: \usepackage{pgfpages}
#+LATEX_HEADER: \mode<handout>
#+LATEX_HEADER: {
#+LATEX_HEADER:   ... see below ...
#+LATEX_HEADER: }
```

- with one frame per A4 page (extending page size)

```
#+LATEX_HEADER:   \pgfpagesuselayout{resize to}[a4paper,landscape]
```

- with two frames per A4 page

```
#+LATEX_HEADER:   \pgfpagesuselayout{2 on 1}[a4paper,border shrink=5mm]
```

- with four frames per A4 page

```
#+LATEX_HEADER:   \pgfpagesuselayout{4 on 1}[a4paper,border shrink=5mm,%
```

Draw a border around the frames

Add a rectangle around each frame in the handout:

```
#+LATEX_HEADER: \setbeamertemplate{background canvas}{
#+LATEX_HEADER:   \tikz \draw (current page.north west) rectangle
#+LATEX_HEADER:                (current page.south east);
#+LATEX_HEADER: }
```

Show speaker notes

Show reminders about what to say during each part of your presentation.
Your laptop monitor and your projector should have the same resolution.
http://freakazoid.teamblind.de/2011/03/30/latex-presentations-with-notes-on-windows-7/

Print handout with speaker notes

See http://tex.stackexchange.com/questions/38084/displaying-slides-with-beamer-and-article-class/38146#38146
See Guido Diepen's handoutWithNotes.sty for PowerPoint like handout.

```
#+LATEX_HEADER: \usepackage{handoutWithNotes}
#+LATEX_HEADER: \pgfpagesuselayout{3 on 1 with notes}[a4paper,border shrink=5mm]
```

Print as article

Using `beamerarticle`.

LaTeX class (XXX not necessary, does work???)

```
#+LATEX_CLASS_OPTIONS:
```

Common options:

- `8pt`, `9pt`, `10pt`, `11pt` (default), `12pt`, `14pt`, `17pt`, `20pt`
- `draft`: no graphics, footlines,. . .
- `handout`: no overlays

```
#+LaTeX_CLASS_options: [bigger,allowframebreaks]
```

Beamer now supports the 16:9 aspect ratio with the `aspectratio` option:

```
#+LaTeX_CLASS_options: [aspectratio=169]
```

LaTeX preamble

Append any line of code in the LaTeX preamble with keywords specific to the LaTeX and Beamer back-ends:

```
#+LATEX_HEADER:       \usepackage{...}
#+LATEX_HEADER_EXTRA: \usepackage{...}
#+BEAMER_HEADER:      \institute[short name]{Institute}
```

It will go (in that order) in the `[EXTRA]` placeholder of the header associated to the `beamer` LaTeX class (see org-latex-classes).

Inline LaTeX I

You can include raw LaTeX in your Org presentations and it will get kept as LaTeX when it's exported.

```
#+begin_LaTeX
\begin{minipage}{4cm}
...
\end{minipage}
#+end_LaTeX
```

```
#+LaTeX: \parbox{4cm}{...}
```

Such LaTeX code will only be present in the exports to LaTeX and Beamer.

```
#+begin_Beamer
\begin{minipage}{4cm}
...
\end{minipage}
#+end_Beamer
```

```
#+Beamer: \parbox{4cm}{...}
```

Such LaTeX code will only be present in the export to Beamer.

Inline LaTeX II

It is especially useful for more advanced stuff like images or tables where you need more control of the LaTeX options than Org mode actually gives you.
For example, to insert a table with colspan or rowspan support:

```
#+begin_LaTeX
\begin{tabular}{|l|l|l|}
  \hline
  Text1 & Text2 & Text3 \\
  \hline
  \multicolumn{3}{|c|}{Merged text here} \\
  \hline
\end{tabular}
#+end_LaTeX
```

LaTeX

Text1	Text2	Text3
Merged text here		

Affiliated keywords

The Beamer back-end reads both

- `#+ATTR_LATEX:` and
- `#+ATTR_BEAMER:`

affiliated keywords.
XXX Code with figure or table

Using a custom theme

You can specify a Beamer theme using the `#+BEAMER_THEME` keyword.
For example:

```
#+BEAMER_THEME: Boadilla
```

which is equivalent (for `Boadilla`) to:

```
#+BEAMER_COLOR_THEME: dolphin
#+BEAMER_FONT_THEME:  default
#+BEAMER_INNER_THEME: [shadow]rounded
#+BEAMER_OUTER_THEME: infolines
```

Changing the frame font

Fonts must be present on the system you're presenting on – or it will go back to a fallback font.
Font Risque.

```
#+LATEX_HEADER: \usepackage[frenchstyle]{kpfonts}
```

Changing font size for example environment

Something along these lines will work:

```
#+LaTeX: {\footnotesize
... Org stuff here ...
#+LaTeX: }
```

Choose the font size you want.
You can also *shrink* individual frames in Beamer by adding a BEAMER option property to the frame's headline. You can also specify a percentage, as in `shrink=10`:

```
* The frame title
  :PROPERTIES:
  :BEAMER_opt: shrink=10
  :END:
```

Adding an image on the title slide

Insert an image in the title slide that fills the whole width of the slide but limits to half height.

```
#+BEAMER_HEADER: \titlegraphic{\includegraphics[width=\textwidth,height=.5\textheig
```

Column view

For a column view of options and configurations for the individual frames

```
#+COLUMNS: %45ITEM %10BEAMER_env(Env) %10BEAMER_act(Act) %4BEAMER_col(Col) %8BEAMER_
#+COLUMNS: %20ITEM %13BEAMER_env(Env) %6BEAMER_envargs(Args) %4BEAMER_col(Col) %7BE
```

Environment specification: BEAMER_env

XXX Put = or ~ around BEAMER_env in title...

- This becomes visible through the `B_frame` tag (visual aid only).

frame

If a heading in the current tree has a `BEAMER_env` property set to either `frame` or `fullframe`, its level overrides the `H` value, giving you some flexibility in deciding what is and what isn't a frame.

```
#+OPTIONS: H:2 toc:t

* Section 1

** Frame

* Section 2

** Subsection 2.1

*** Frame                                                    :B_frame:
    :PROPERTIES:
    :BEAMER_env: frame
    :END:

This becomes a frame, instead of a block!
```

This works in both "directions": to add or to remove sectioning levels above the current heading (which becomes

If a heading in the current tree has a `BEAMER_env` property set to `fullframe`, the frame will not display its title (`frametitle` is being set to the empty string).

```
*** Frame                                                    :B_fullframe:
    :PROPERTIES:
    :BEAMER_env: fullframe
    :END:

This becomes a frame, with its title ignored!
```

`block` environment

Insert a default block (syntax assumes `H:2`).

```
*** A block

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.
```

A block

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Comments:

- Use the `BEAMER_env` property to specify a different block type for the current "block" environment.
- Use ignoreheading to terminate a block environment

`alertblock` environment

Insert a block whose title is highlighted.

```
*** An alert block                                                :B_alertblock:
    :PROPERTIES:
    :BEAMER_env: alertblock
    :END:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.
```

An alert block

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

exampleblock environment

Insert a block that is supposed to be an example.

```
*** An example block                                          :B_exampleblock:
    :PROPERTIES:
    :BEAMER_env: exampleblock
    :END:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.
```

An example block

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

`beamercolorbox` environment

Create colored boxes whose "beamer color" to use (= a pair of colors: `bg` + `fg`) is defined by the heading text, for example `title in head/foot` or `myblockcolor`.

```
#+LaTeX: \setbeamercolor{myblockcolor}{bg=magenta,fg=white}

*** myblockcolor                                          :B_beamercolorbox:
    :PROPERTIES:
    :BEAMER_env: beamercolorbox
    :END:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

beamercolorbox environment

```
*** myblockcolor                                              :B_beamercolorbox:
    :PROPERTIES:
    :BEAMER_env: beamercolorbox
    :BEAMER_opt: wd=6cm
    :END:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

beamercolorbox environment

```
*** myblockcolor                                              :B_beamercolorbox:
    :PROPERTIES:
    :BEAMER_env: beamercolorbox
    :BEAMER_opt: shadow=false,rounded=true
    :END:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

beamercolorbox environment

```
*** myblockcolor                                              :B_beamercolorbox:
    :PROPERTIES:
    :BEAMER_env: beamercolorbox
    :BEAMER_opt: shadow=true,rounded=true
    :END:

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua.
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

`structureenv` environment

- For highlighting text.
- To help the audience see the structure of your presentation.

Paragraph Heading. Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

`definition` environment

- Behaves like the `theorem` environment, except that the theorem style `definition` is used.
- In this style, the body of a `theorem` is typeset in an upright font.

```
*** Prime number                                                    :B_definition:
    :PROPERTIES:
    :BEAMER_env: definition
    :END:

A prime number is a natural number greater than 1 that has no positive divisors
other than 1 and itself.
```

Definition (Prime number)

A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.

`example` environment

- Behave like the `theorem` environment, except that the theorem style `example` is used.
- A side-effect of using this theorem style is that the contents is put in an `exampleblock` instead of a `block`.

```
*** Number 5                                                    :B_example:
    :PROPERTIES:
    :BEAMER_env: example
    :END:

5 can only be divided evenly by 1 or 5, so it is a prime number.
```

Example (Number 5)

5 can only be divided evenly by 1 or 5, so it is a prime number.

`theorem` environment

- Insert a theorem.
- Italicized body.

```
*** Niessen, 2010                                                    :B_theorem:
    :PROPERTIES:
    :BEAMER_env: theorem
    :END:

Statement of theorem (proposition): \\
Org mode + Beamer = productivity^{2}
```

Theorem (Niessen, 2010)

Statement of theorem (proposition):
Org mode + Beamer = productivity^2

`proof` environment

- Typeset a proof.
- Period (full-stop) added at the end of the title.
- Control sequence `\qed` (*quod erat demonstrandum*, for the traditional "tombstone") added at the end of the proof.

```
*** Proof                                                          :B_proof:
    :PROPERTIES:
    :BEAMER_env: proof
    :END:

Description of proof.
```

Proof.

Description of proof. □

`quote` environment

Use `quote` to typeset short quoted text (or a series of small quotes, separated by blank lines): it hasn't paragraph indentation.

```
#+begin_quote
"Don't give up on your dreams, keep on sleeping." --- Albert Einstein
#+end_quote
```

> *"Don't give up on your dreams, keep on sleeping."*
> — *Albert Einstein*

If you want to place the source or the author's name at the right, after the quote, you can use these commands:

```
#+LaTeX: \begin{raggedleft}
(Albert Einstein)
#+LaTeX: \par\end{raggedleft}
```

(Albert Einstein)

`quotation` environment

Use `quotation` to typeset longer quoted text (one or more paragraphs) because it indents the first line of each paragraph.

```
#+begin_quotation
"Two things are infinite, the universe and human stupidity, and I am not yet
completely sure about the universe."
--- Albert Einstein
#+end_quotation
```

> *"Two things are infinite, the universe and human stupidity, and I am not yet completely sure about the universe."* — *Albert Einstein*

`verse` environment

Use `verse` for quotations where line breaks are important, such as poetry.

```
#+begin_verse
"Sit next to a pretty girl for an hour,
it seems like a minute.
Sit on a red-hot stove for a minute,
it seems like an hour.
That's relativity!"
--- Albert Einstein
#+end_verse
```

> *"Sit next to a pretty girl for an hour,*
> *it seems like a minute.*
> *Sit on a red-hot stove for a minute,*
> *it seems like an hour.*
> *That's relativity!"*
> *— Albert Einstein*

Alternative environment syntax (without option)

- More readable, more portable when environments exist (LaTeX + other backends)

```
#+begin_theorem
There is no largest prime number.
#+end_theorem
```

- More powerful: you can't nest blocks of the same type with this syntax, and more portable when environments only exist in Beamer (then, they will appear as subheadings when exported to other backends)
- Use an empty heading (or `\phantom{dummy}`) for an empty block title.

```
***                                                          :B_theorem:
    :PROPERTIES:
    :BEAMER_env: theorem
    :END:
```

Alternative environment syntax (with option)

```
#+attr_latex: :options [Lagrange]
#+begin_theorem
Let $G$ be a finite group, and let $H$ be a subgroup of $G$.  Then the order of
$H$ divides the order of $G$.
#+end_theorem
```

Is the syntax with [...] or with {...}?

```
*** Lagrange                                                        :B_theorem:
    :PROPERTIES:
    :BEAMER_env: theorem
    :END:

Let $G$ be a finite group, and let $H$ be a subgroup of $G$.  Then the order of
$H$ divides the order of $G$.
```

Alternative environment syntax (with parameter)

XXX How to pass the parameter?

Definition

Prime number A prime number is

Definition (Prime number)

A prime number is

```
#+attr_latex: :options {Prime number}
#+begin_definition
A prime number is
#+end_definition
```

```
*** Prime number                                              :B_definition:
    :PROPERTIES:
    :BEAMER_env: definition
    :END:

A prime number is
```

Add extra environments

For simple environments, use:
I think we should changes some environment placeholders:

- Introduce %r which would stand for the raw heading (without any processing)
- %H and %U would use the raw heading text instead.

The previous definition would become:
WDYT?
It never would have occurred to me on my own to use the heading text for LaTeX code. That's a conceptual leap that passed me by.

- Environment options may be given using the BEAMER_opt property. They will be enclosed in square brackets and inserted where %o appears in the environment definition. (with an example, but I can't think of one now)
- Additional arguments may be written into the

Column specification: BEAMER_COL

Splitting a frame into multiple columns

To get multiple columns in a frame:

1. Press `C-c C-b |` (`BMCOL`) on the headings (inside the frame) which will become columns
   The heading of column environments won't be outputted in the PDF file.
2. Specify the column width as a percentage of `\textwidth`
   No absolute width (such as `4cm`), which wouldn't be correctly translated...

Instead of `block`, those structural environments will become `column` (with the width parameter as a factor of `\textwidth`). Consecutive `column` environments will be put in a `columns` environment.

First column
Two \ lines.
The arithmetic mean is equal to the

Second column
One line (but aligned).
$\frac{1}{n}\sum_{i=1}^{n} x_i$

column

You can change the percent space of each column.
If you want like one column to take 70% and the other to take 30%, you can change that as follows:
XXX

,label=sec:org7e69142]`columns` Colonne 1.
Colonne 2.

How to specify the best width?

Visualise the width by transforming the columns into blocks. . .

Result of an evaluation on two columns

Balancing text in columns.

- Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
- Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

- Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
- Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

appendix

You can add an appendix (frames that you do not intend to show during your talk, but which might be useful to answer a question) by inserting such a level 1 heading after the last regular slide of your actual presentation:

```
* Appendix material follows                                    :B_appendix:
  :PROPERTIES:
  :BEAMER_env: appendix
  :END:

# Backup slides
```

Ignoring page number in backup slides can be achieved by setting the option `noframenumbering` on all "backup" slides.

note

To keep your presentation notes (reminders about what to say), add a heading below the frame, and select the `note` environment with `C-c C-b`.

```
#+LATEX_HEADER: \setbeameroption{show notes}

*** Some note                                                      :B_note:
    :PROPERTIES:
    :BEAMER_env: note
    :END:

This is a note.

- Stress this first.
- Then this.
```

noteNH

Note with its title ignored.

againframe

You can "continue" frames that you previously started somewhere (but where certain details have been suppressed) at a much later point (for example only in the appendix) to show additional slides there.
For convenience (since those are mandatory), when asking for an `againframe`, Org Beamer always asks for:

- `BEAMER_ref` property to refer, using link syntax, to (the label of) the frame being resumed, and
- `BEAMER_act` property to set the overlay specification.

An advantage is that you don't need to know the label of the frame being resumed: `:BEAMER_ref: *My frame title`.
Contents are ignored.
XXX How to label a frame? Via `:BEAMER_opt: label=id` or via other means?

ignoreheading

Need to "close" a block environment, if followed by something else than a block.

```
*** End of block                                                :B_ignoreheading:
    :PROPERTIES:
    :BEAMER_env: ignoreheading
    :END:
```

- A heading with an `ignoreheading` environment will have only its contents inserted in the output.
  - Contents is not inserted in any `frame` environment...
- This special value is useful to have data between frames, or to properly close a `column` environment.

Overlay specification: BEAMER_act

Set overlay / action specifications in current frame or block to create dynamic effects (*multiple slides*, called *overlays*, for a single frame) = old `BEAMER_envargs` property.
Overlay specifications are given inside angular brackets.
XXX <> seem to be added when they aren't present.
Copied as is if present.
Headings support the `BEAMER_act` property:

```
** Heading
   :PROPERTIES:
   :BEAMER_act: [<+->]
   :END:

- Item
- Item
```

It is translated as:

- an overlay/action specification, or
- a default overlay specification when enclosed within square brackets.

Overlay specification: BEAMER_act

Set overlay / action specifications in current frame or block to create dynamic effects (*multiple slides*, called *overlays*, for a single frame) = old `BEAMER_envargs` property.
Overlay specifications are given inside angular brackets.
XXX <> seem to be added when they aren't present.
Copied as is if present.
Headings support the `BEAMER_act` property:

```
** Heading
   :PROPERTIES:
   :BEAMER_act: [<+->]
   :END:

- Item
- Item
```

It is translated as:

- an overlay/action specification, or
- a default overlay specification when enclosed within square brackets.

Overlay specification: BEAMER_act

Set overlay / action specifications in current frame or block to create dynamic effects (*multiple slides*, called *overlays*, for a single frame) = old `BEAMER_envargs` property.
Overlay specifications are given inside angular brackets.
XXX <> seem to be added when they aren't present.
Copied as is if present.
Headings support the `BEAMER_act` property:

```
** Heading
   :PROPERTIES:
   :BEAMER_act: [<+->]
   :END:

- Item
- Item
```

It is translated as:

- an overlay/action specification, or
- a default overlay specification when enclosed within square brackets.

Overprint

> What may not be easy or possible is to use the directive, which is > what I used in my previous response to you.
You can always use the only environment. https://github.com/suvayu/.emacs.d/blob/master/org-mode-config.el#L215
That said, I think overlays with only is not as smooth as with simple overlay specifications to regular environments or macros like `\includegraphics`, `\item`, etc.

Overlay specification: BEAMER_act

The Queen's old armchair

- Princess Anne
- Prince Charles
- corgis

Question on ML

The following works for me:

```
#+Beamer: \only<1>{
[[file:figure1.png]]
#+Beamer: }\only<2>{
[[file:figure2.png]]
#+Beamer: }\only<3->{
[[file:figure3.png]]
#+Beamer: }
```

There is the `BEAMER_act` property that can be used to apply overlay information on blocks but I don't think it's possible on individual figures. Of course, you could put each figure in a separate block. The following/attached will match what you had originally.

```
#+options: H:1
* The slide
** figure 1
   :PROPERTIES:
   :beamer_act: <1>
   :END:
   [[file:chromosome.png]]
** figure 2
```

Option specification: BEAMER_opt

Insert optional arguments for the current `frame` or `block` environment using the `BEAMER_OPT` property.
The options will automatically be enclosed within square brackets. Don't enclose them yourself!
You can add that special property by editing the `Opt` column within the "column view":

1. Press `C-c C-x C-c`
2. Go to the `Opt` column
3. Press `e` to edit its contents
4. Quit XXX

`fragile` option is added automatically.

Vertical alignment

You can specify *top* vertical alignment globally by the `t` class option:

```
#+LaTeX_CLASS_OPTIONS: [t]
```

For single frames, you can use the same option locally:

```
** Vertically top-aligned
   :PROPERTIES:
   :BEAMER_opt: t
   :END:

Some content.
```

Explicit page breaking

If the text does not fit on a single slide, all you have to do to automatically break up the frame into several frames, is set the option `allowframebreaks`.

- To allow frame breaks on a frame by frame basis[2]:

```
** A very long "frame" with breaks
   :PROPERTIES:
   :BEAMER_opt: allowframebreaks,label=
   :END:
```

  XXX This property shouldn't be interpreted for the current slide!
- You might want to put `allowframebreaks=0.9` there
- To add an explicit page break:

```
#+beamer: \framebreak
```

- To allow frame breaks for the whole document[3]:

```
#+BIND: org-beamer-frame-default-options "allowframebreaks"
```

[2] Until the Beamer issue #265 is solved, we need to unset the

Frame without surroundings

`plain` causes the headlines, footlines, and sidebars to be suppressed. This is useful for creating single frames with different head- and footlines or for creating frames showing big pictures that completely fill the frame.

```
**                                                  :some_tag_so_that_heading_is_empty:
   :PROPERTIES:
   :BEAMER_opt: plain
   :END:

#+begin_beamer
\begin{centering}
\includegraphics[height=\textheight]{somebigimagefile}
\par\end{centering}
#+end_beamer
```

:some_tag_so_that_heading_is_empty:

Big title in the middle

This is my BIG title!

Skip proof

Proof details

Summary

For further reading

A. Salomaa.
Formal Languages.
Academic Press, 1973.

For further reading

A. Salomaa.
Formal Languages.
Academic Press, 1973.

E. Dijkstra.
Smoothsort, an alternative for sorting in situ.
Science of Computer Programming, 1(3):223–233, 1982.

For further reading

A. Salomaa.
Formal Languages.
Academic Press, 1973.

E. Dijkstra.
Smoothsort, an alternative for sorting in situ.
Science of Computer Programming, 1(3):223–233, 1982.

E. Feldman and J. Owings, Jr.
A class of universal linear bounded automata.
Information Sciences, 6:187–190, 1973.

For further reading


A. Salomaa.
Formal Languages.
Academic Press, 1973.


E. Dijkstra.
Smoothsort, an alternative for sorting in situ.
Science of Computer Programming, 1(3):223–233, 1982.


E. Feldman and J. Owings, Jr.
A class of universal linear bounded automata.
Information Sciences, 6:187–190, 1973.


P. Jančar, F. Mráz, M. Plátek, and J. Vogel.
Restarting automata.
FCT Conference 1995, LNCS 985, pages 282–292. 1995.

Proof details

Text omitted in main talk.

More details

Even more additional material.

Abbreviations

Issues

Report issues and suggest features and improvements on the GitHub issue tracker.

Patches

I love contributions! Patches under any form are always welcome!

Donations

If you use the refcard-org-beamer project and feel it is making your life better and easier, you can show your appreciation and help support future development by making a donation through PayPal. Thank you!
Regardless of the donations, refcard-org-beamer will always be free both as in beer and as in speech.

Author: Fabrice Niessen
Keywords: reference card org-beamer