Recipes for presentations with beamer latex using emacs org-mode

Arne Babenhauserheide

August 8, 2012

Outline

Usage

- (configure your emacs, see Basic Configuration at the end)
- C-f <file which ends in .org>
- Insert heading:

```
Hello World

#+LaTeX_CLASS: beamer
#+BEAMER_FRAME_LEVEL: 2

* Hello
** Hello GNU
Nice to see you!
```

- M-x org-export-as-pdf

  done: Your first org-beamer presentation.

org-mode + beamer = love

- Code

```
Recipes
#+LaTeX_CLASS: beamer
#+BEAMER_FRAME_LEVEL: 2
* Introduction
** org-mode + beamer =  love
*** Code :BMCOL:
    :PROPERTIES:
    :BEAMER_col: 0.7
    :END:
<example block>
*** Simple block  :BMCOL:B_block:
    :PROPERTIES:
    :BEAMER_col: 0.3
    :BEAMER_env: block
    :END:
it's that easy!
```

Simple block

it's that easy!

Two columns - in commands

Commands

```
** Two columns - in commands
*** Commands
C-c C-b | 0.7
C-c C-b b
C-n
<eTAB (write example) C-n C-n
*** Result
C-c C-b | 0.3
C-c C-b b
even easier - and faster!
```

Result

even easier - and faster!

Four blocks - code

```
*** Column 1 :B_ignoreheading:BMCOL:
    :PROPERTIES:
    :BEAMER_env: ignoreheading
    :BEAMER_col: 0.5
    :END:

*** One
*** Three

*** Column 2 :BMCOL:B_ignoreheading:
    :PROPERTIES:
    :BEAMER_col: 0.5
    :BEAMER_env: ignoreheading
    :END:

*** Two
*** Four
```

Four blocks - result

- One
- Three

- Two
- Four

Four nice blocks - commands

```
***
C-c C-b | 0.5 # column
C-c C-b i # ignore heading
*** One
C-c C-b b # block
*** Three
C-c C-b b
***
C-c C-b | 0.5
C-c C-b i
*** Two
C-c C-b b
*** Four
C-c C-b b
```

Four nice blocks - result

One

Two

Three

Four

Top-aligned blocks

Code

```
*** Code
    :PROPERTIES:
    :BEAMER_env: block
    :BEAMER_col: 0.5
    :BEAMER_envargs: C[t]
    :END:

*** Result
    :PROPERTIES:
    :BEAMER_env: block
    :BEAMER_col: 0.5
    :END:
pretty nice!
```

Result

pretty nice!

Two columns with text underneath - code

- Code

```
***  :B_columns:
    :PROPERTIES:
    :BEAMER_env: columns
    :END:

**** Code :BMCOL:
    :PROPERTIES:
    :BEAMER_col: 0.6
    :END:

**** Result :BMCOL:
    :PROPERTIES:
    :BEAMER_col: 0.4
    :END:

*** Underneath :B_ignoreheading:
    :PROPERTIES:
    :BEAMER_env: ignoreheading
    :END:
Much text underneath! Very Much.
Maybe too much. The whole width!
```

- Result

Much text underneath! Very Much. Maybe too much. The whole width!

Nice quotes

Code

```
#+begin_quote
Emacs org-mode is a
great presentation tool -
Fast to beautiful slides.
- Arne Babenhauserheide
#+end_quote
```

Result

> *Emacs org-mode is a great presentation tool - Fast to beautiful slides.*
>
> - *Arne Babenhauserheide*

Math snippet

Code

Inline

```
\( 1 + 2 = 3 \) is clear
```

As equation

```
\[ 1 + 2 \cdot 3 = 7 \]
```

Result

Inline

$1 + 2 = 3$ is clear

As equation

$$1 + 2 \cdot 3 = 7$$

LaTeX

Code

```
\( \LaTeX \) gives a space
after math mode.

\LaTeX{} does it, too.

\LaTeX does not.

At the end of a sentence
both work.
Try \LaTeX. Or try \LaTeX{}.

Only \( \LaTeX \) and \( \LaTeX{} \)
also work with HTML export.
```

Result

LaTeX gives a space after math mode.
LaTeX does it, too.
LaTeXdoes not.
At the end of a sentence both work. Try LaTeX. Or try LaTeX.
Only LaTeX and LaTeX also work with HTML export.

Images with caption and label

Code

```
#+caption: GNU Emacs icon
#+label: fig:emacs-icon
[[/usr/share/icons/hicolor/128x128/a
```

```
This is image (\ref{fig:emacs-icon})
```

Result

Figure : GNU Emacs icon

This is image (1)

Autoscaled to the block width!

Examples

Code

```
: #+bla: foo
: * Example Header
```

Gives an example, which does not interfere with regular org-mode parsing.

```
#+begin_example
content
#+end_example
```

Gives a simpler multiline example which *can* interfere.

Result

```
#+bla: foo
* Example Header
```

Gives an example, which does not interfere with regular org-mode parsing.

```
content
```

Gives a simpler multiline example which *can* interfere.

Header

```
<Title>

#+startup: beamer
#+LaTeX_CLASS: beamer
#+LaTeX_CLASS_OPTIONS: [bigger]
#+AUTHOR: <empty for none, if missing: inferred>
#+DATE: <empty for none, if missing: today>
#+BEAMER_FRAME_LEVEL: 2
#+TITLE: <causes <Title> to be regular content!>
```

.emacs config

Put these lines into your .emacs or in a file your .emacs pulls in - i.e. via *(require 'mysettings)* if the other file is named *mysettings.el* and ends in *(provide 'mysettings)*.

```
(org-babel-do-load-languages ; babel, for executing
 'org-babel-load-languages   ; code in org-mode.
 '((sh . t)
   (emacs-lisp . t)))

(require 'org-latex) ; latex export
(add-to-list         ; with highlighting
  'org-export-latex-packages-alist '("" "minted"))
(add-to-list
  'org-export-latex-packages-alist '("" "color"))
(setq org-export-latex-listings 'minted)
```

.emacs variables

You can easily set these via *M-x customize-variable*.

```
(custom-set-variables ; in ~/.emacs, only one instance
 '(org-export-latex-classes (quote ; in the init file!
    (("beamer" "\\documentclass{beamer}"
       org-beamer-sectioning))))
 '(org-latex-to-pdf-process (quote
    ((concat "pdflatex -interaction nonstopmode"
             "-shell-escape -output-directory %o %f")
     "bibtex $(basename %b)"
     (concat "pdflatex -interaction nonstopmode"
             "-shell-escape -output-directory %o %f")
     (concat "pdflatex -interaction nonstopmode"
             "-shell-escape -output-directory %o %f")))))
```

(concat "..." "...") is used here to get nice, short lines. Use the concatenated string instead ("pdflatex...%f").

Required programs

- Emacs - (gnu.org/software/emacs)

  To get org-mode and edit .org files effortlessly.

  ```
  emerge emacs
  ```

- Beamer LaTeX - (bitbucket.org/rivanvx/beamer)

  To create the presentation.

  ```
  emerge dev-tex/latex-beamer app-text/texlive
  ```

- Pygments - (pygments.org)

  To color the source code (with minted).

  ```
  emerge dev-python/pygments
  ```

Thanks

Thanks go to the writers of emacs and org-mode, and for this guide in particular to the authors of the org-beamer tutorial on worg.
Thank you for your great work!

[1]: As additional permission under GNU GPL version 3 section 7, you may distribute these works without the copy of the GNU GPL normally required by section 4, provided you include a license notice and a URL through which recipients can access the Corresponding Source and the copy of the GNU GPL.