The wiki2beamer example

October 23, 2018

Welcome

Welcome to the wiki2beamer example presentation. We will do our best to document and demonstrate all features of wiki2beamer in this file. It is not meant to be a reference though, that will be the man page wiki2beamer(1).

Design goals

Wiki2beamer was written to make typesetting presentations with LaTeXand beamer easier. LaTeXbeamer can create beautiful presentations but it's a very verbose language and most of the simple tasks like using frames or bullet lists take an awful amount of code that looks really complicated. This often scares away beginners and fellow colleagues from even starting with LaTeX. This is where wiki2beamer steps in. It's designed to make the start easier and not remove any of the powers of the language.

Commandline usage

Wiki2beamer is written in python and known to work on python 2.3 up to 2.6. You can use it on Windows too, if you have the python interpreter installed properly.
On *nix environments you would simple use it like this:

```
1 wiki2beamer example.txt > example.tex
```

On Windows you would have to call the python interpreter with the wiki2beamer script:

```
1 python wiki2beamer example.txt > example.tex
```

The big picture

A latex beamer document consists (like any latex document) of some introductory style definitions and the document body. In plain latex it looks like this:

```
\documentclass{beamer}
    %do some style-magic here
\begin{document}
    %insert your presentation here (body)
\end{document}
```

wiki2beamer can produce both: only the body content of the presentation or a complete document.

body-only

If you want full control over all the fancy stuff you can do with latex, you can use wiki2beamer to generate a .tex that that you can include with \input{filename.tex}.

```
\documentclass{beamer}
\begin{document}
    \input{w2b-output.tex}
\end{document}
```

This is the default mode of operation.

autotemplate

If you want wiki2beamer to generate a complete document with header, you can use the autotemplate feature. It's a special environment in wiki2beamer syntax and must be the first element in the input file. It has a simple `key={value}` syntax. For a full reference see the man page.

```
<[autotemplate]
title={The Title}
author={Mr. Foo Bar}
date={1970-01-01}
titleframe=False
[autotemplate]>
```

If the autotemplate environment is used, a default titleframe is generated. If you want that disabled, you can switch it with the titleframe option.

Structure and frames

As most wiki-dialects, wiki2beamer supports the famous

```
==== title ====
```

syntax. When these markings start at the beginning of a line, wiki2beamer expands them to open a frame.

Structure and frames

The following heading markups exist:

`== title ==` → opens a section

`=== title ===` → opens a subsection

`==== title ====` → opens a frame

Frames will be closed automatically when a new sectioning or frame markup appears. Frames can also be closed with an optional `[frame]>` marking (usually only needed to write advanced LaTeX code between frames).

Lists

Probably one of the most used text elements in presentations are:

- bullet
- lists

and

1. numbered
2. lists

Bullet Lists

Bullet lists can be created by prepending one or many * before a line

```
* A simple bullet list
* with two items
```

Bullet list can also be multi-level:

```
* A more complex bullet list
** with a sublist
** and another sublist
*** here
```

Bullet Lists (output)

- A simple bullet list
- with two items

- A more complex bullet list
  - with a sublist
  - and another sublist
    - here

Numbered Lists

The same works for numbered lists

```
# first item
# second item
## with two subitems
## and another
### subsubitem
# third item
```

Numbered Lists (output)

1. first item
2. second item
   - 2.1 with two subitems
   - 2.2 and another
     - 2.2.1 subsubitem
3. third item

Mixed Lists

Numbered and bullet lists can be mixed:

```
# first item
#* with a sub-list
#*# one numbered subsub-item
#*# another numbered subsub-item
#* and a lone bullet
# second item
```

Mixed Lists (output)

1. first item
   - with a sub-list
     1.1 one numbered subsub-item
     1.2 another numbered subsub-item
   - and a lone bullet
2. second item

Text markup

Text can be formatted with some markups. Some markups can be escaped with a \.

markup	output	escaping
`'''bold text'''`	**bold text**	
`''italic text''`	*italic text*	
`@teletype text@`	`teletype text`	\@
`!alerted text!`	alerted text	\!
`_color_colored text_`	blue text	

Text substitutions

There are some simple text replacements you can use:

markup	output	escaping
`-->`	→	
`<--`	←	
`==>`	⇒	
`<==`	⇐	
`:-)`	☺	
`:-(`	☹	

Footnotes

You can also use footnotes on your slides[1].

```
...your slides(((This is such a footnote.))).
```

[1] This is such a footnote.

Environments

One of the basic building blocks of LaTeX syntax are environments. In plain LaTeX you open, give options and close them with the following notation:

```
\begin{someenv}[option=foo]
...
\end{someenv}
```

wiki2beamer provides a shorter syntax that looks like this:

```
<[someenv][option=foo]
...
[someenv]>
```

Environment example

Some of the most used environments probably is:

```
<[center] some centered text [center]>
```

output:

some centered text

LaTeX vs. wiki2beamer environments

The default case is that wiki2beamer doesn't care about the name of the environment and just passes it on to LaTeX. For these default environments wiki2beamer doesn't track open and close tags.

There are a few exceptions where wiki2beamer knows and parses the environments:

- `[nowiki]` (escaping from wiki2beamer)
- `[code]` (code listings)
- `[autotemplate]` (autotemplate header)

Escaping from wiki2beamer

Everything wiki2beamer doesn't know, it doesn't touch. But what if it knows something that you want it to leave untouched?
For some of the most common notations we already have escaping with a \. If there isn't an escape, you can use the generic [nowiki] environment.

```
1 <[nowiki]
2 '''text that isn't bold'''
3 [nowiki]>
```

Notice that the [nowiki] tags start and end at the beginning of a line.

Escaping (output)

"'text that isn't bold"'

Code

One of the great strengths of LaTeX is the ability to typeset and highlight sourcecode. Doing that manually is a very tedious task in visual presentation tools. In LaTeX code listings are realized with the `listings` package that provides a new `\lstlisting` environment.

In wiki2beamer we have a convenient `[code]` environment.

Code

The basic usage is:

```
<[code]
... your listing here ...
[code]>
```

The `lstlisting` environment in the background provides options, e.g. you can configure the highlighting:

```
<[code][style=basic,language=C,title=code example]
if ( a == b ) { return 0; }
[code]>
```

Inside [code] environments, wiki2beamer processing is mostly disabled. The only characters with a special meaning are [and] which have to be escaped with a \.

Code (output)

code example

```
1 if ( a == b ) { return 0; }
```

Vertical Space

You can insert vertical space between two paragraphs with the

`--length--`

notation where *length* can be a valid latex length expression (e.g. `1cm`, `1pt`, `1em` or even `0.2\textwidth`). This space will be squeezed by latex when the page is full. To really force LaTeX to insert a space, use the alternative

`--*length--`

notation.

Columns

Latex beamer can divide the current frame into columns. In wiki2beamer this can be done with the following notation:

```
<[columns]

[[[width]]]
this is the first column

[[[width]]]
this is the second column

[columns]>
```

where *width* is a valid latex length expression.

Columns example

```
\centering
Text above the columns

<[columns]

[[[0.4\textwidth]]]
this is:
* the first column
* with a list

[[[0.4\textwidth]]]
this is:
* the second column
* with another list

[columns]>

\centering
Text below the columns
```

Columns example (output)

Text above the columns.

this is:

- the first column
- with a list

this is:

- the second column
- with another list

Text below the columns.

Headers and Footers

Sometimes you want to repeat some latex code at the header and footer of your presentation frames (e.g. to show logos or names). To make this easier you can fill two special storage field `@FRAMEHEADER=` and `@FRAMEFOOTER=` with some latex code. wiki2beamer will append or prepend this right after the frame opening or frame closing in the resulting latex code.

Frame with generated header/footer

This is a generated FRAMEHEADER

This is the text of the frame

This is a generated FRAMEFOOTER

Including graphics

LaTeX can include graphics with the `\includegraphics` command. To ease the use, wiki2beamer provides a simple syntax.
simple:

```
<<<graphicsfile.png>>>
```

advanced:

```
<<<graphicsfile.png,option=foo>>>
```

For a documentation of available options, take a look at the graphix package documentation.

Graphics example

```
<<<db-rg1024.png,height=0.5\textheight>>>
```

Graphics example (output)

Figure Example

Alternatively you could use the `figure` environment which would allow you to add a `caption` too:

```
<[figure]
<<<db-rg1024.png,height=0.5\textheight>>>
\caption{Database Scheme}
<[figure]
```

Figure Example (output)

Figure: Database Scheme

Animation

LaTeX beamer provides facilities for generating simple animated slides. Wiki2beamer provides some shortcuts on top of the beamer class.

LaTeX beamer has the notion of animation layers: Whenever a frame contains an animation, it will consist of consecutive numbered animation layers.

Layer specs

Whenever you animate something, you can specify that it should appear or disappear on a certain animation layer. The notations can look like this:

`<n>`	a single layer *n*
`<n,m>`	two layers *n* and *m*
`<n-m>`	all layers between *n* and *m*
`<n-m,k>`	all layers between *n* and *m* plus layer *k*

Animating lists

To animate a wiki2beamer list, just add a spec after the * or # characters:

```
* always there (layer 1-4)
*<2> only on layer 2
*<2-3> on layer 2-3
*<2,4> on layer 2 and 4
```

Animating lists (output)

- always there (layer 1-4)

Animating lists (output)

- always there (layer 1-4)
- only on layer 2
- on layer 2-3
- on layer 2 and 4

Animating lists (output)

- always there (layer 1-4)
- on layer 2-3

Animating lists (output)

- always there (layer 1-4)
- on layer 2 and 4

Animating anything

LaTeX knows the two commands \uncover and \only.

uncover shows and element on the given layers while otherwise just displaying an invisible placeholder box. In wiki2beamer this is denoted as:

+<n-m>{content}

here's an example:

```
1 above
2
3 +<2>{uncovered}
4
5 below
```

`uncover` example (output)

above

below

`uncover` example (output)

above
uncovered
below

only example

only makes an element appear without having a placeholder box before. In wiki2beamer this is denoted as:

-<n-m>{content}

here's the example:

```
1 above
2
3 -<2>{only on 2}
4
5 below
```

`only` example (output)

above
below

`only` example (output)

above
only on 2
below

animating graphics

Animating graphics is nothing special. Here is the example:

```
+<1>{
        <<<db-rg1024.png,width=0.4\textwidth>>>
}
+<2>{
        <<<db-rg1024.png,width=0.4\textwidth>>>
}
```

animating graphics (output)

animating graphics (output)

Animating code

When talking about code listings, you often want things to appear or disappear, step by step. In wiki2beamer, you can do that with a simple notation[2].

```
<[code]
some normal code above
[<2-4>some code only on layer 2-4]
[[<2>some code on layer 2][<4>replaced on layer 4]]
some normal code below
[code]>
```

[2] Try to do it without wiki2beamer, you'll feel the pain ;)

Animating code (output)

```
1 some normal code above
2
3
4 some normal code below
```

Animating code (output)

```
1 some normal code above
2 some code only on layer 2-4
3 some code on layer 2
4 some normal code below
```

Animating code (output)

```
1 some normal code above
2 some code only on layer 2-4
3
4 some normal code below
```

Animating code (output)

```
1 some normal code above
2 some code only on layer 2-4
3 replaced on layer 4
4 some normal code below
```

Advanced usage

Tricks for the adept.

Frame options

LaTeX supports giving arguments to opening frames. It looks like:

```
\begin{frame}[someoption]
```

Such options can be added in wiki2beamer by appending them to the frame opening:

```
==== frametitle ====[someoption]
```

One example are fragile frames.

Fragile frames

For some features to work (eg. verbatim content), latex beamer needs a frame marked as fragile. In plain LaTeXthis would look like:

```
\begin{frame}[fragile]
...
\end{frame}
```

This can be realized with wiki2beamer by simply appending a `[fragile]` tag to the frame header:

```
==== a fragile frame ====[fragile]
...
```

a fragile frame

This is such a fragile frame with some `V#rb/\t|m t<xt` .

Short section names

The same notation that applies for frame options can be used with sections. This is used to assign a short name of the section for the table of contents.

```
1 === some section name that is mich too long ===[long section name]
```

Selective editing

When LaTeX documents become large and contain many inputs (graphics), the latex compiler tends to be slow – to slow for the edit-compile cycle.

To speed things up a bit, wiki2beamer allows you to select single frames for editing.

```
!==== frame title ====
```

When there is at least one selected frame, all frames that are not selected will be omitted from the LaTeX output.

Managing input

If you want to work with your colleagues on a single presentation you can (and should) of course use your favourite version control system, but you'd still have to resolve conflicts. To reduce the number of conflicts, wiki2beamer can build presentations from multiple input files.

Multiple inputs by commandline

The first way is to use the commandline to concatenate multiple input files:

```
1 wiki2beamer header.txt dave.txt debby.txt > talk.tex
```

Multiple inputs by includes

The second way is to include additional input files with the

```
>>>includefile.txt<<<
```

syntax.