

The dynblocks package *

Claudio Fiandrino
claudio.fiandrino@gmail.com

September 20, 2012

Abstract

The dynblocks package allows to fully customize blocks aspect and dimension inside a presentation.

The original idea for this package came from [this question](#) in [TeX.SE](#) and the core of the package is based on [this answer](#).

Contents

1	Introduction	1
2	Usage	2
2.1	Basic usage	2
2.2	Change text alignment	4
2.3	Text opacity and word alert	4
3	Options and advanced examples	6
3.1	The shadow and the rounded corners	7
3.2	Customized fill colors	8
3.3	Color adaptation to the Beamer theme	10

1 Introduction

The purpose of the package is to have an instrument to customize several aspects of blocks (here called *dynblocks*):

- the width;
- the color
 - of the background;

*This package has version number *v0.2* of 20/09/2012

- of the border;
- the text:
 - alignment;
 - opacity

Notice that *dynblocks* defined by *dynblocks* differ mainly from usual beamer's blocks because no title is given.

The package has the following requirements:

- TikZ;
- etoolbox;
- xparse.

2 Usage

To load the package use as usual: `\usepackage[⟨options⟩]{dynblocks}`.

The different options that can be adopted will be analysed in detail in section 3.

2.1 Basic usage

Using the package in basic mode allows to define a block with:

- *justified* alignment;
- width equal to `\textwidth`;
- border color `\blue` and fill color `\blue!10`.

thanks to the command `\opaqueblock<⟨overlay spec⟩>[⟨width⟩]{⟨text⟩}`. Moreover, it is also possible to make it *invisible* forcing colors to become gray by means of `\invblock<⟨overlay spec⟩>`. For example, the following code, generates the two frames shown in figures [1a](#) and [1b](#).

```
\documentclass{beamer}
\usepackage{dynblocks}

\usetheme{Luebeck}

\begin{document}
\begin{frame}{The frame title}
\begin{columns}[T]
\begin{column}{0.4\textwidth}
\begin{dynblock}
\opaqueblock<1>[0.8\textwidth]{hello this is a dynamic block
```

```

with an itemize environment:
\begin{itemize}
  \item hello
  \item hello again
\end{itemize}
}
\invblock<2->
\end{dynblock}
\end{column}
\begin{column}{0.4\textwidth}
  \begin{dynblock}
    \opaqueblock<2>{hello this is another dynamic block}
  \end{dynblock}
\end{column}
\end{columns}
\end{frame}

\end{document}

```

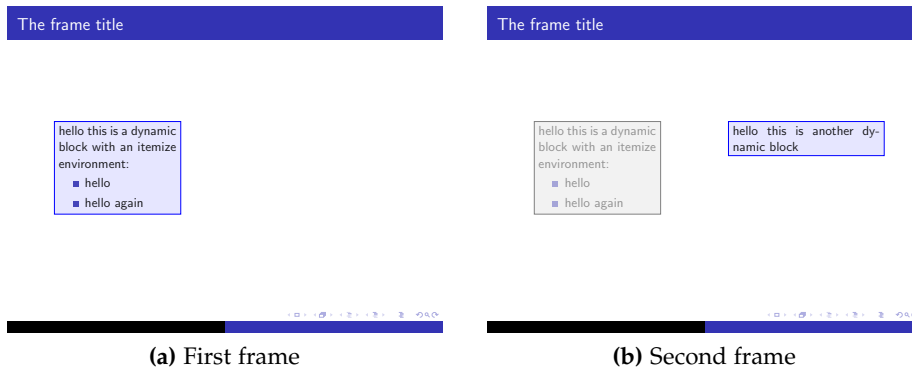


Figure 1: The basic example

In this example, it is possible to notice that the second `\opaqueblock` has not width specified; the default value is `\textwidth`, but if the block is placed inside a *column* environment it inherits the width given there. To set different values of width, it is necessary to specify the optional argument as did for the first `\opaqueblock` of the example.

The presence of an `\invblock` makes the first `\opaqueblock` invisible; this command needs to be placed immediately after an `\opaqueblock` because of two facts:

- it automatically inherits the width from the `\opaqueblock`;
- it automatically shows the text of the previous `\opaqueblock`.

Finally, both `\opaqueblock` and `\invblock` needs to be placed inside a `dynblock` environment.

2.2 Change text alignment

Thanks to the command `\setalignment{⟨text spec⟩}` the text alignment changes according to the `⟨text spec⟩` (for all possibilities please refer to the pgfmanual section 16.4.3 - version October 25, 2010).

Suppose, for instance, to modify the previous example in order to display the second block with center aligned text. The only change to do is:

```
\begin{dynblock}  
\setalignment{center}  
\opaqueblock<2>{hello this is another dynamic block}  
\end{dynblock}
```

and this will lead to figure 2.

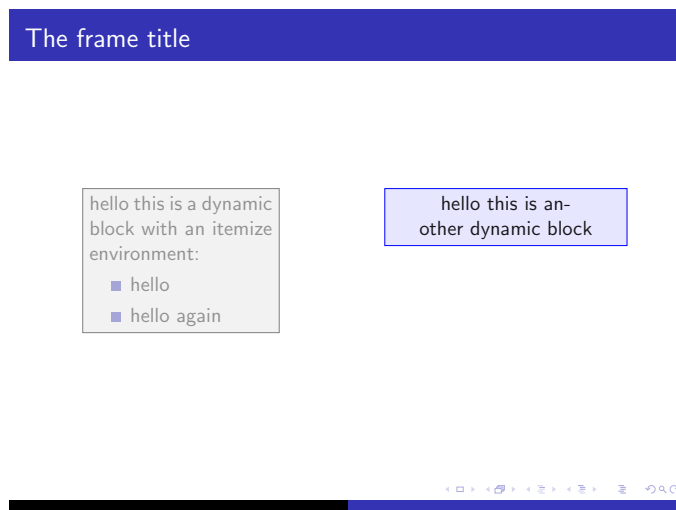


Figure 2: Second frame with text center aligned for the second block

Notice that alignments can be local or global: if there is a definition in the preamble, perfectly admissible, all *dynblocks* will be set according to this definition. Whereas the definition is set inside a group, the *columns* environment or simply the *dynblock* environment, it affects just local *dynblocks*. It is even possible set a global definition and then change things locally.

2.3 Text opacity and word alert

By default:

- `\opaqueblocks` have an opacity set to 0.9;
- `\invblocks` have an opacity set to 0.4.

To change these values two commands have been introduced:

- `\setvisopacity{⟨opacity spec⟩};`
- `\setinvopacity{⟨opacity spec⟩};`

where $\langle \text{opacity spec} \rangle$ is a value in the interval $[0, 1]$. Also these commands can be set locally or in a global fashion in the preamble.

Due to the opacity the usual `\alert` command is not more useful with `\invblocks`. The package provides a method to alert a word even in this case; the command to be used is `\dynalert<⟨overlay spec⟩>{⟨text⟩}`; notice that through the proper usage of $\langle \text{overlay spec} \rangle$, `\dynalert` must not fall inside a `\opaqueblock`. This is a limitation, but the purpose for which it has been developed is different.

Assume, for example, to modify the reference example such that the opacity of `\invblock` will be set to 0.1; furthermore, for the first block, the word “*itemize*” will be alerted with `\alert`, while “*dynamic block*” with `\dynalert` and for the second block (that does not have the correspondent `\invblock`) it is shown what happens with a wrong usage of `\dynalert`. The code is:

```
\documentclass{beamer}
\usepackage{dynblocks}

\usetheme{Luebeck}
\setinvopacity{0.1}

\begin{document}
\begin{frame}{The frame title}
\begin{columns}[T]
\begin{column}{0.4\textwidth}
\begin{dynblock}
\opaqueblock<1>[0.8\textwidth]{hello this is a
\dynalert<2>{dynamic block} with an \alert<1,2>{itemize}
environment:
\begin{itemize}
\item hello
\item hello again
\end{itemize}
}
\invblock<2->
\end{dynblock}
\end{column}
\begin{column}{0.4\textwidth}
\begin{dynblock}
\opaqueblock<2>{hello this is another
\dynalert<2>{dynamic} block}
\end{dynblock}
\end{column}
\end{columns}
```

```
\end{frame}

\end{document}
```

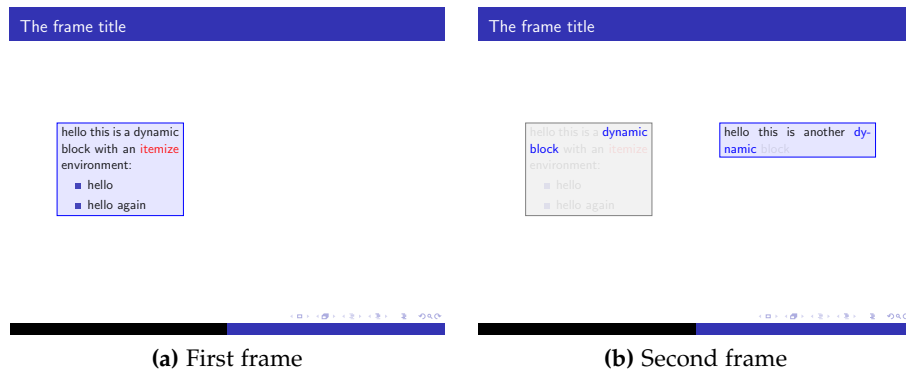


Figure 3: Example with different opacity and alerts

As it is possible to see from figures 3a and 3b, the usual `\alert`, when used inside an `\invblock`, is set with the opacity of the block while the proper `\dynalert` no. Anyway, a wrong usage of `\dynalert` lead to the output shown in figure 3b: the subsequent text of the alerted word is set with the opacity of an `\invblock`.

The suggested use, in conclusion, is:

- `\alert` with $\langle overlay spec \rangle$ equal specified in the related `\opaqueblock`;
- `\dynalert` to highlight words inside an `\invblock`;
- never do something like: `\dynalert<1,2>\{word\}` if the `\opaqueblock` is shown in $\langle overlay spec \rangle = 1$ and the `\invblock` in $\langle overlay spec \rangle = 2$.

To change colors:

- for `\alert` the usual Beamer command works:
`\setbeamercolor{alerted text}{\langle color spec \rangle};`
- for `\dynalert` a different command has been introduced to differentiate them from standard Beamer's alerts: `\setwordscolor{\langle color \rangle};` the default value is set to *blue*.

3 Options and advanced examples

In this section the package's options are introduced with examples. They allow to customize more deeply the aspect of *dynblocks*:

- adding the shadow and the rounded corners (subsection 3.1);
- customizing the fill color (subsection 3.2);
- adapting the fill color to the current Beamer theme used (subsection 3.3).

3.1 The shadow and the rounded corners

To load:

- the shadow option use: `\usepackage[shadow]{dynblocks}`; it is possible to set the *shadow opacity* by means of the following command: `\setshadowopacity{<opacity spec>}` (default value 0.4);
- the option to have rounded corners for *dynblocks* use: `\usepackage[roundedcorners]{dynblocks}`.

For example:

```

\documentclass{beamer}
\usepackage[shadow,roundedcorners]{dynblocks}

\usetheme{Luebeck}

\begin{document}
\begin{frame}{The frame title}
\begin{columns}[T]
\begin{column}{0.4\textwidth}
\begin{dynblock}
\opaqueblock<1>[0.8\textwidth]{hello this is a
\dynalert<2>{dynamic block} with an
\alert<1,2>{itemize} environment:
\begin{itemize}
\item hello
\item hello again
\end{itemize}
}
\invblock<2->
\end{dynblock}
\end{column}
\begin{column}{0.4\textwidth}
\begin{dynblock}
\setalignment{center}
\opaqueblock<2>{hello this is another dynamic block}
\end{dynblock}
\end{column}
\end{columns}
\end{frame}

\end{document}

```

allows to get the frames shown in figures 4a and 4b.

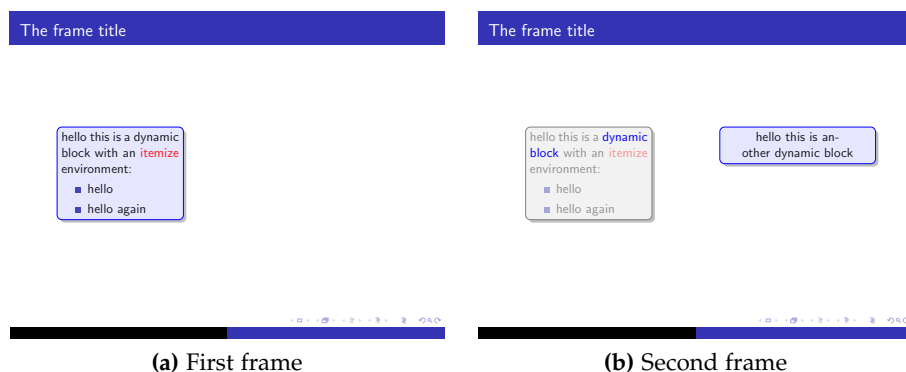


Figure 4: Example with shadow and roundedcorners options

3.2 Customized fill colors

By activating this option it is possible to fully customize the *dynblocks* colors because several command become available:

- `\setblockcolor{<color spec>}` and `\setbordercolor{<color spec>}` for the `\opaqueblocks` (default values are *blue!10* and *blue* respectively);
- `\setinnercolor{<color spec>}` and `\setoutercolor{<color spec>}` for the `\fancyblocks` (default values are *white* and *blue!10* respectively);
- `\settopcolor{<color spec>}` and `\setbottomcolor{<color spec>}` for the `\vshadeblocks` (default values are *white* and *blue!10* respectively);
- `\setleftcolor{<color spec>}` and `\setrightcolor{<color spec>}` for the `\oshadeblocks` (default values are *white* and *blue!10* respectively).

Similarly to the shadow and the roundedcorners options, to load the customcolors option use `\usepackage[customcolors]{dynblocks}`.

In the following example, all *dynblocks* types are used and it is possible to see how local and global setting work.

```

\documentclass{beamer}
\usepackage[shadow, roundedcorners, customcolors]{dynblocks}
% some global settings
\setblockcolor{red!10}
\setbordercolor{red}
\setbottomcolor{orange!40}
\setrightcolor{orange!40}

\usetheme{Luebeck}

\begin{document}
\begin{frame}{The frame title}
\begin{columns}[T]

```



```

\begin{column}{0.4\textwidth}
\begin{dynblock}
\opaqueblock<1>[0.8\textwidth]{hello this is a
\dynalert<2>{dynamic block} with an
\alert<1,2>{itemize} environment:
\begin{itemize}
\item hello
\item hello again
\end{itemize}
}
\invblock<2->
\end{dynblock}
\end{column}
\begin{column}{0.4\textwidth}
\setalignment{center}
\begin{dynblock}
% default settings since no
% \setinnercolor or \setoutercolor
% are there
\fancyblock<2>{hello this is another dynamic block}
\invblock<3->
\end{dynblock}
\\[2ex]
\setbordercolor{orange} % local definition
% that overwrites the global one
\begin{dynblock}
\vshadeblock<3>{replica: hello this a another dynamic block}
\invblock<4->
\end{dynblock}
\\[2ex]
\begin{dynblock}
\oshadeblock<4>{replica 2: hello this a another dynamic block}
\end{dynblock}
\end{column}
\end{columns}
\end{frame}

\end{document}

```

The four frames obtained by this example are shown in figures [5a](#), [5b](#), [5c](#) and [5d](#); more in detail:

- an example of `\opaqueblock` customization through `\setblockcolor` and `\setbordercolor` could be seen in figure [5a](#);
- an example of `\fancyblock` with default settings could be seen in figure [5b](#) (notice that it inherits the `bordercolor` from the global setting);
- an example of `\vshadeblock` with customization of `\setbottomcolor` and locally `\setbordercolor` (`\settopcolor` at default value) could be seen in figure [5c](#);

- finally, an example of `\oshadeblock` with `\setrightcolor` and local `\setbordercolor` customization (`\settopcolor` at default value) could be seen in figure 5d.



Figure 5: Example with `customcolor` option and all *dynblocks* types

3.3 Color adaptation to the Beamer theme

The purpose of this option is to use the Beamer's color of the theme currently adopted; as it will be possible to see, the `getthemecolors` option should be used with particular care. To load the option there is the usual `\usepackage[getthemecolors]{dynblocks}`.

This option is defined inside the package as:

```
\DeclareOption{getthemecolors}{
% redefinition opaqueblock
\renewcommand{\thecol}{structure.fg!10}
\renewcommand{\thebordercol}{structure.fg}
% redefinition fancyblock
\def\@setinnercolor{white}
\def\@setoutercolor{structure.fg!10}
% redefinition vshadeblock
```

```

\def\@settopcolor{white}
\def\@setbottomcolor{structure.fg!10}
% redefinition oshadeblock
\def\@setleftcolor{white}
\def\@setrightcolor{structure.fg!10}
}

```

thus it works properly if the current beamercolortheme set the *structure* definition.

For example:

```

\documentclass{beamer}
\usepackage[shadow, roundedcorners, getthemecolors,
  customcolors]{dynblocks}

\usetheme{CambridgeUS}

\begin{document}
\begin{frame}{The frame title}
\begin{columns}[T]
\begin{column}{0.4\textwidth}
\begin{dynblock}
\opaqueblock<1>[0.8\textwidth]{hello this is a
\dynalert<2>{dynamic block} with an
\alert<1,2>{itemize} environment:
\begin{itemize}
\item hello
\item hello again
\end{itemize}
}
\invblock<2->
\end{dynblock}
\end{column}
\begin{column}{0.4\textwidth}
\setalignment{center}
\begin{dynblock}
\opaqueblock<2>{hello this is another dynamic block}
\end{dynblock}
\end{column}
\end{columns}
\end{frame}

\end{document}

```

will lead to figures [6a](#) and [6b](#).

The result can be improved in the following way:

```

\documentclass{beamer}
\usepackage[shadow, roundedcorners, getthemecolors,
  customcolors]{dynblocks}

\usetheme{CambridgeUS}
% definition of structure

```

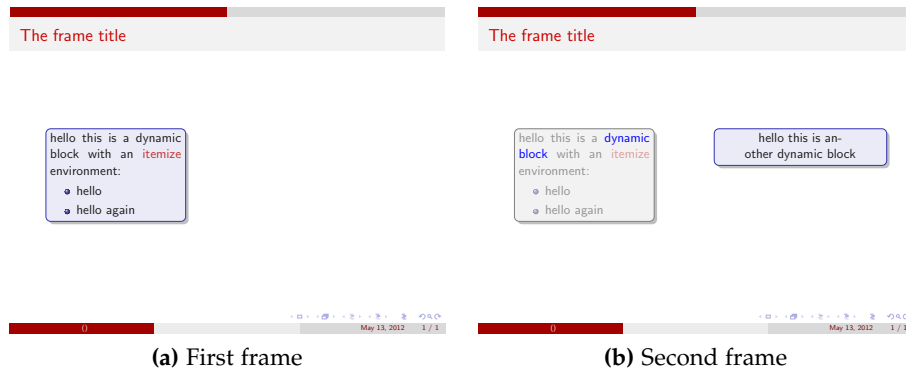


Figure 6: Example with a theme that does not define structure color

```
\setbeamercolor*{structure}{parent=palette primary}

\begin{document}
\begin{frame}{The frame title}
\begin{columns}[T]
\begin{column}{0.4\textwidth}
\begin{dynblock}
\opaqueblock<1>[0.8\textwidth]{hello this is a
\dynalert<2>{dynamic block} with an
>alert<1,2>{itemize} environment:
\begin{itemize}
\item hello
\item hello again
\end{itemize}
}
\invblock<2->
\end{dynblock}
\end{column}
\begin{column}{0.4\textwidth}
\setalignment{center}
\begin{dynblock}
\opaqueblock<2>{hello this is another dynamic block}
\end{dynblock}
\end{column}
\end{columns}
\end{frame}

\end{document}
```

obtaining as result the frames shown in figures 7a and 7b.

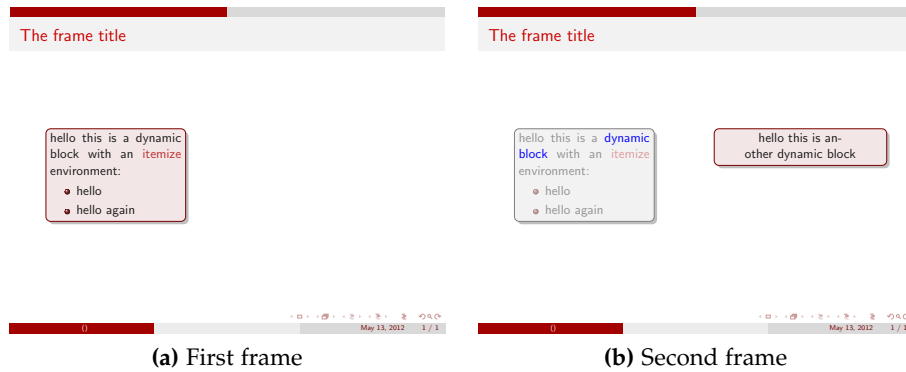


Figure 7: Example of a theme with a posteriori structure color definition

Here is another example:

```

\documentclass{beamer}
\usepackage[getthemecolors]{dynblocks}

\usetheme{EastLansing}
\setbeamercolor*{structure}{parent=palette primary}

\begin{document}
\begin{frame}{The frame title}
\begin{dynblock}
\opaqueblock<1>{hello this is a
dynamic block with an
itemize environment:
\begin{itemize}
\item hello
\item hello again
\end{itemize}
}
\end{dynblock}
\end{frame}

\end{document}

```

with the final output shown in figure 8.

The decision of adopting the *structure* as reference is due to the fact that this parameter assume an high relevance when customizing a theme. In the following example, it is shown a color customization of the Szeged theme and a particular effect that can be realized thanks to multiple *dynblocks* inside the same dynblock environment:

```

\documentclass{beamer}
\usepackage[getthemecolors,roundedcorners,shadow]{dynblocks}

\usetheme{Szeged}

```

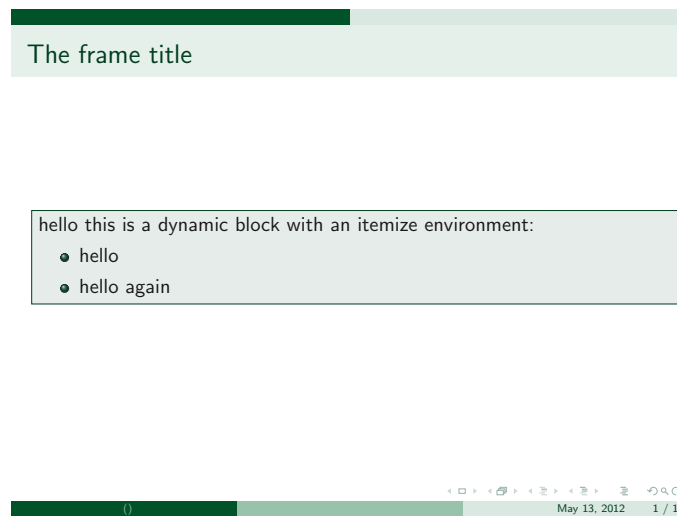


Figure 8: Second example of a theme with a posteriori structure color definition

```
\setbeamercolor{structure}{bg=red!20,fg=red}

\begin{document}

\begin{frame}{A title}
\begin{center}
\begin{dynblock}
\opaqueblock<1>[0.6\textwidth]{hello this is a dynamic block
  with an itemize environment:
\begin{itemize}
\item hello
\item hello again
\end{itemize}
}
\invblock<2->
\setalignment{center}
\opaqueblock<2>{hello this is another dynamic block}
\end{dynblock}
\end{center}
\end{frame}

\end{document}
```

The two frames obtained as outcome are shown in figures [9a](#) and [9b](#).

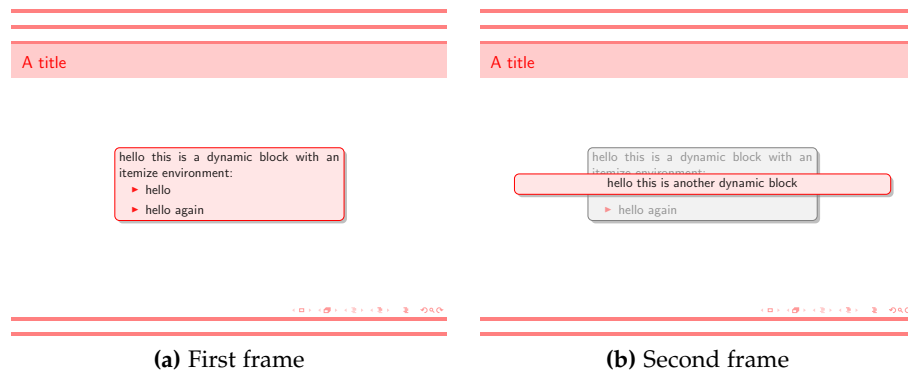


Figure 9: Example of a customized theme

Notice that the `getthemecolors` option has some drawbacks when used with particular Beamer color themes like:

- albatross;
- beetle.