

LaTeX Beamer Class による穴埋め式 handout/slide

樋口さぶろお Saburo Higuchi

龍谷大学理工学部数理情報学科

2012

目標

- 1 LaTeX Beamer Class で PowerPoint より簡単に、領域の非表示/表示の変化をつけよう (アニメーションはありません)
- 2 LaTeX Beamer Class で もんだメソッドを使おう.
- 3 LaTeX Beamer Class で空欄に書き込める handout を作ろう



<http://hig3.net>

Commands

- `\hiddenbox{ 隠す部分 }`
このように指定された「隠す部分」が、前から1個ずつ表示される
- `\hiddenbox< 指定 >{ 隠す部分 }`
「隠す部分」が表示される部分を、前から順、でなく、指定 (overlay specification) の通りとする。この指定方法は LaTeX Beamer Class が定めている。

Some syntactic sugar

- `\hiddenboxm{ 隠す部分 }, \hiddenboxm< 指定 >{ 隠す部分 }`
数式の中で使える。 `hiddenbox{$ $}` のこと。
- `\hiddenpar{ 隠す部分 }, \hiddenpar< 指定 >{ 隠す部分 }`
隠す部分、がスライド幅の minipage にはいつているものとして、minipage 全体を表示/非表示する (いってみれば、高さが保たれる)。黒板で説明することをメモしてほしいとき (スライドを作る時間が足りないとき?) は、`\hiddenpar{\ \vspace*{5cm}}` などとしておくと、メモ必要性が伝わりやすい。

以下使用例

もともと LaTeX Beamer には, `overlay` という機能がある.

`beameruserguide.pdf` の `Creating Overlays` という section を参照.

今の目的には, 非表示時にも同じ面積を占有し (すなわち `only` ではなく), また薄く表示されるのではなく (すなわち `cover/uncover` でなく), 完全に無地になる `visible/invisible` が適している. ここで定義した `hiddenbox` は `visible` をちょっと修飾したもの.

次は, デフォルトで `itemize` に適用された `cover/uncover` の例

- 1 A
- 2 B
- 3 C
- 4 D

以下使用例

もともと LaTeX Beamer には, `overlay` という機能がある.

`beameruserguide.pdf` の `Creating Overlays` という section を参照.

今の目的には, 非表示時にも同じ面積を占有し (すなわち `only` ではなく), また薄く表示されるのではなく (すなわち `cover/uncover` でなく), 完全に無地になる `visible/invisible` が適している. ここで定義した `hiddenbox` は `visible` をちょっと修飾したもの.

次は, デフォルトで `itemize` に適用された `cover/uncover` の例

- 1 A
- 2 B
- 3 C
- 4 D

以下使用例

もともと LaTeX Beamer には, `overlay` という機能がある.

`beameruserguide.pdf` の `Creating Overlays` という section を参照.

今の目的には, 非表示時にも同じ面積を占有し (すなわち `only` ではなく), また薄く表示されるのではなく (すなわち `cover/uncover` でなく), 完全に無地になる `visible/invisible` が適している. ここで定義した `hiddenbox` は `visible` をちょっと修飾したもの.

次は, デフォルトで `itemize` に適用された `cover/uncover` の例

- ① A
- ② B
- ③ C
- ④ D

以下使用例

もともと LaTeX Beamer には, `overlay` という機能がある.

`beameruserguide.pdf` の `Creating Overlays` という section を参照.

今の目的には, 非表示時にも同じ面積を占有し (すなわち `only` ではなく), また薄く表示されるのではなく (すなわち `cover/uncover` でなく), 完全に無地になる `visible/invisible` が適している. ここで定義した `hiddenbox` は `visible` をちょっと修飾したもの.

次は, デフォルトで `itemize` に適用された `cover/uncover` の例

- 1 A
- 2 B
- 3 C
- 4 D

以下使用例

もともと LaTeX Beamer には, `overlay` という機能がある.

`beameruserguide.pdf` の `Creating Overlays` という section を参照.

今の目的には, 非表示時にも同じ面積を占有し (すなわち `only` ではなく), また薄く表示されるのではなく (すなわち `cover/uncover` でなく), 完全に無地になる `visible/invisible` が適している. ここで定義した `hiddenbox` は `visible` をちょっと修飾したもの.

次は, デフォルトで `itemize` に適用された `cover/uncover` の例

- 1 A
- 2 B
- 3 C
- 4 D

hiddenbox の使用

オイラーの公式を発見したのは

テイラー展開を発見したのは

$$e^{i\theta} = \text{\texttt{hiddenbox}} + \sqrt{-1} \text{\texttt{hiddenbox}}$$

hiddenbox の使用

オイラーの公式を発見したのは Euler

テイラー展開を発見したのは

$$e^{i\theta} = \text{} + \sqrt{-1} \text{}$$

hiddenbox の使用

オイラーの公式を発見したのは Euler

テイラー展開を発見したのは Taylor

$$e^{i\theta} = \boxed{} + \sqrt{-1} \boxed{}$$

hiddenbox の使用

オイラーの公式を発見したのは Euler

テイラー展開を発見したのは Taylor

$$e^{i\theta} = \cos \theta + \sqrt{-1} \boxed{}$$

hiddenbox の使用

オイラーの公式を発見したのは Euler

テイラー展開を発見したのは Taylor

$$e^{i\theta} = \cos \theta + \sqrt{-1} \sin \theta$$

hiddenbox の制限

$$\boxed{} e^x = x$$

$$\boxed{} \log x = x$$

$$\boxed{} = e^{A+B}$$

以前は align や $\boxed{}$ など, 内部的に複数回評価されるところではうまく動かなかったんだけど, 今は大丈夫.

$\boxed{}$ 側で bug fix されたのか?

hiddenbox の制限

$$\log e^x = x$$

$$\log x = x$$

$$= e^{A+B}$$

以前は align や $\log x = x$ など、内部的に複数回評価されるとうまく動かなかっただけで、今は大丈夫.

$\log x = x$ 側で bug fix されたのか?

hiddenbox の制限

$$\log e^x = x$$

$$\exp \log x = x$$

$$\boxed{\phantom{e^{A+B}}} = e^{A+B}$$

以前は align や $\boxed{\phantom{e^{A+B}}}$ など、内部的に複数回評価されるとうまく動かなかっただけで、今は大丈夫.

$\boxed{\phantom{e^{A+B}}}$ 側で bug fix されたのか?

hiddenbox の制限

$$\log e^x = x$$

$$\exp \log x = x$$

$$e^A e^B = e^{A+B}$$

以前は align や など、内部的に複数回評価されるとうまく動かなかっただけ、今は大丈夫。

側で bug fix されたのか？

hiddenbox の制限

$$\log e^x = x$$

$$\exp \log x = x$$

$$e^A e^B = e^{A+B}$$

以前は align や `eqnarray` など、内部的に複数回評価されるとうまく動かなかっただけ、今は大丈夫。

側で bug fix されたのか?

hiddenbox の制限

$$\log e^x = x$$

$$\exp \log x = x$$

$$e^A e^B = e^{A+B}$$

以前は align や `eqnarray` など、内部的に複数回評価されるとうまく動かなかっただけ、今は大丈夫.

`LaTeX Beamer Class` 側で bug fix されたのか?

表示順序の制御とデフォルトの overlay との連携

<> を使って書く場合は, その <> の中はすべて手で指定. 書式は LaTeX Beamer Class の beameruserguide.pdf の Creating Overlays の section 参照.



- ① D
- ② E
- ③ F
- ④ G

表示順序の制御とデフォルトの overlay との連携

<> を使って書く場合は, その <> の中はすべて手で指定. 書式は LaTeX Beamer Class の beameruserguide.pdf の Creating Overlays の section 参照.



- ① D
- ② E
- ③ F
- ④ G

表示順序の制御とデフォルトの overlay との連携

<> を使って書く場合は, その <> の中はすべて手で指定. 書式は LaTeX Beamer Class の beameruserguide.pdf の Creating Overlays の section 参照.

A **B** **C**

- ① D
- ② E
- ③ F
- ④ G

表示順序の制御とデフォルトの overlay との連携

<> を使って書く場合は, その <> の中はすべて手で指定. 書式は LaTeX Beamer Class の beameruserguide.pdf の Creating Overlays の section 参照.



- ① D
- ② E
- ③ F
- ④ G

表示順序の制御とデフォルトの overlay との連携

<> を使って書く場合は, その <> の中はすべて手で指定. 書式は LaTeX Beamer Class の beameruserguide.pdf の Creating Overlays の section 参照.



- ① D
- ② E
- ③ F
- ④ G

有限の高さを持つ領域の uncover

$$\cosh x = \frac{1}{2}(e^{+x} + e^{-x})$$

有限の高さを持つ領域の uncover

$$e^x = \sum_{n=0}^{\infty} \frac{1}{n!} x^n$$

LaTeX Beamer Class

<https://bitbucket.org/rivanvx/beamer/wiki/Home>

Main Features(引用)

- The list of features supported by Beamer is quite long (unfortunately, so is presumably the list of bugs supported by Beamer). The most important features, in our opinion, are:
- You can use Beamer both with pdflatex and latex+dvips.
- The standard commands of LaTeX still work. A will still create a table of contents, section is still used to create structure, and itemize still creates a list.
- You can easily create overlays and dynamic effects.
- Themes allow you to change the appearance of your presentation to suit you purposes.
- The themes are designed to be usable in practice, they are not just for show. You will not find such nonsense as a green body text on a picture of a green meadow.
- The layout, the colors, and the fonts used in a presentation can easily be changed globally, but you still also have control over the most minute detail.
- A special style file allows you to use the LaTeX source of a presentation directly in other LaTeX classes like article or book. This makes it easy to create presentations out of lecture notes or lecture notes out of presentations.
- The final output is typically a PDF file. Viewer applications for this format exist for virtually every platform. When bringing your presentation to a conference on a memory stick, you do not have to worry about which version of the presentation program might be installed there. Also, your presentation is going to look exactly the way it looked on your computer.

もんだメソッド

<http://d.hatena.ne.jp/keyword/もんだメソッド>