LaTeX : Introduction à Beamer

Dorian Depriester

29 janvier 2012

Source disponible à la page :
http://blog.dorian-depriester.fr/?p=235

Présentation des blocks

Qui servent à encadrer les éléments

Le principe de Beamer est basé sur l'utilisation des blocks, qui portent un nom et un contenu.

Un block normal

Beamer c'est facile

Block d'exemple (*exampleblock*)

La preuve !

Bloc d'alerte (*alertblock*)

Penser à fermer l'environnement *block*

Théorème

1+1=2

Corollaire

2+2=4

Démonstration.

C'est trivial □

Remarques

Vous aurez relevé l'intérêt de tels environnements. Il existe aussi *lemma* et *example*. Très pratiques pour les matheux.
La traduction –de "theorem" en "théorème" par exemple– est assurée par les arguments donnés plus haut :

```
\uselanguage{French}
\languagepath{French}
```

L'environnement *columns*

Pour mettre cote à cote du texte, des images ou des blocks, il faut commencer par dire que l'on va séparer l'espace en plusieurs colonnes (*columns*), puis on spécifie chaque colonne (*columns*).

Le bloc de gauche

Regardez à droite

Le code

```
\begin{columns}
  \begin{column}{0.35\textwidth}
    \begin{block}{Le bloc de gauche}
      Regardez à droite
    \end{block}
  \end{column}
  \begin{column}{0.65\textwidth}
    \begin{block}{Le code}
      ...
    \end{block}
  \end{column}
\end{columns}
```

Remarque

Vous aurez compris qu'il vaut mieux bien marquer les retraits dans le code si on veut pas s'y perdre dans les différents environnements.

Syntaxe

Le plus simple pour afficher progressivement des éléments est de préciser entre chevrons (<n>) à quel(s) overlay(s) on veut que tel élement s'affiche.

Numérotation des overlays

Il est bien sûr possible d'afficher un élément sur plusieurs overlays, auquel cas on sépare leurs numéros par des virgules, ou par un tiret (par exemple <1-3> pour dire de 1 à 3), ou encore <2-> pour dire du deuxième au dernier.

Un block habituel

- Toto

Un block habituel

- Toto
- Tata

Un block habituel

- Toto
- Tata
- Tata et toto

Un block habituel

- Toto
- Tata

- Titi

Un block habituel

- Toto
- Tata
- Titi

Avec les blocks

Ça marche aussi !

onslide$< n >$ { Très pratique pour afficher les colonnes d'un tableau les unes après les autres (à mettre en argument dans la définition du tableau, au moment où on indique les filets verticaux et les alignements).

Automatisation des overlays

Comme illustré ci-dessus, on peut automatiser les overlays dans certains environnements, tels que *itemize* et *description*, en ajoutant l'argument $[< +- >]$ à l'ouverture de l'environnement.

onslide< n > { Très pratique pour afficher les colonnes d'un tableau les unes après les autres (à mettre en argument dans la définition du tableau, au moment où on indique les filets verticaux et les alignements).

only< n > { Tous les éléments dans le *only* ne seront présents qu'aux overlays spécifiés, sans prendre de place s'ils sont cachés.

Automatisation des overlays

Comme illustré ci-dessus, on peut automatiser les overlays dans certains environnements, tels que *itemize* et *description*, en ajoutant l'argument [< +− >] à l'ouverture de l'environnement.

onslide< n > { Très pratique pour afficher les colonnes d'un tableau les unes après les autres (à mettre en argument dans la définition du tableau, au moment où on indique les filets verticaux et les alignements).

only< n > { Tous les éléments dans le *only* ne seront présents qu'aux overlays spécifiés, sans prendre de place s'ils sont cachés.

visible< n > { Tous les éléments dans le *visible* ne seront visibles qu'aux overlays spécifiés, l'espace est occupé s'ils sont cachés.

Automatisation des overlays

Comme illustré ci-dessus, on peut automatiser les overlays dans certains environnements, tels que *itemize* et *description*, en ajoutant l'argument [< +− >] à l'ouverture de l'environnement.

onslide< n > { Très pratique pour afficher les colonnes d'un tableau les unes après les autres (à mettre en argument dans la définition du tableau, au moment où on indique les filets verticaux et les alignements).

only< n > { Tous les éléments dans le *only* ne seront présents qu'aux overlays spécifiés, sans prendre de place s'ils sont cachés.

visible< n > { Tous les éléments dans le *visible* ne seront visibles qu'aux overlays spécifiés, l'espace est occupé s'ils sont cachés.

invisible< n > { L'inverse, logique.

Automatisation des overlays

Comme illustré ci-dessus, on peut automatiser les overlays dans certains environnements, tels que *itemize* et *description*, en ajoutant l'argument [< +− >] à l'ouverture de l'environnement.

onslide$< n >\{$ Très pratique pour afficher les colonnes d'un tableau les unes après les autres (à mettre en argument dans la définition du tableau, au moment où on indique les filets verticaux et les alignements).

only$< n >\{$ Tous les éléments dans le *only* ne seront présents qu'aux overlays spécifiés, sans prendre de place s'ils sont cachés.

visible$< n >\{$ Tous les éléments dans le *visible* ne seront visibles qu'aux overlays spécifiés, l'espace est occupé s'ils sont cachés.

invisible$< n >\{$ L'inverse, logique.

alt$< n >\{x\}\{y\}$ Propose un texte alternatif. Aux slides autres que n, x sera affiché, ils sera remplacé par y à n.

Automatisation des overlays

Comme illustré ci-dessus, on peut automatiser les overlays dans certains environnements, tels que *itemize* et *description*, en ajoutant l'argument $[< +- >]$ à l'ouverture de l'environnement.

Mise en évidence d'un texte

La commande *alert* permet de mettre en évidence du texte :

- Un item pas intéressant
- Un autre qui sert à rien
- Celui qui vaut vraiment le coup
- Un dernier pour la route

Mise en évidence d'un texte

La commande *alert* permet de mettre en évidence du texte :

- Un item pas intéressant
- Un autre qui sert à rien
- Celui qui vaut vraiment le coup
- Un dernier pour la route

Mise en évidence d'un texte

La commande *alert* permet de mettre en évidence du texte :

- Un item pas intéressant
- Un autre qui sert à rien
- Celui qui vaut vraiment le coup
- Un dernier pour la route

Modification du style

- un qui va devenir gras
- un qui va devenir italique
- un qui va devenir bleu
- un qui va être écrit avec serif

Mise en évidence d'un texte

La commande *alert* permet de mettre en évidence du texte :

- Un item pas intéressant
- Un autre qui sert à rien
- Celui qui vaut vraiment le coup
- Un dernier pour la route

Modification du style

- un qui va devenir **gras**
- un qui va devenir *italique*
- un qui va devenir bleu
- un qui va être écrit avec serif

Automatiser *alert*

Comme pour l'apparition, on peut automatiser la surbrillance successive des items d'une liste avec [<alert@+ >] :

1. On va parler d'abord de ça,
2. puis de ça,
3. et enfin de ce truc.

Automatiser *alert*

Comme pour l'apparition, on peut automatiser la surbrillance successive des items d'une liste avec [<alert@+ >] :

1. On va parler d'abord de ça,
2. puis de ça,
3. et enfin de ce truc.

Automatiser *alert*

Comme pour l'apparition, on peut automatiser la surbrillance successive des items d'une liste avec [<alert@+ >] :

1. On va parler d'abord de ça,
2. puis de ça,
3. et enfin de ce truc.

Créer des liens entre slides

Grâce au package *hyperref* (chargé automatiquement pas Beamer), il est possible de créer des liens vers n'importe quel label.

Exemples de liens

- Cliquez ici si vous n'avez pas saisi comment faire des animations
- Exemple de bouton généré par beamer : Retour sur les animations