

## TD N° 2

### Supports de présentation orale et compléments pour vos documents

## La classe BEAMER

La classe de document `beamer` permet de créer rapidement des présentations de qualité pour vidéoprojecteur ou rétroprojecteur. Cette classe est disponible sur le serveur `uranie` avec une documentation très complète :

`/usr/local/share/texmf-local/tex/latex/beamer/doc/beameruserguide.pdf`

### Prise en main

**1.** Créez un nouveau répertoire et copiez le fichier `~ponsini/LaTeX/supports.tex`. Éditez ce fichier avec EMACS. Notez les commandes qui définissent les paramètres du document, celles qui permettent d'ajouter des « cadres<sup>1</sup> » et celles qui permettent de composer les cadres de titre et de sommaire.

**2.** Compilez le fichier source (deux fois) avec `latex`. Le document produit n'est pas directement visible avec `xdvi`, il faut le convertir en fichier *PostScript* avec la commande `dvips`. Vous pouvez maintenant le visualiser avec `gv`.

**3.** La classe BEAMER génère automatiquement des liens hypertextes dans le document. Pour en profiter pleinement, convertissez votre document au format PDF avec la commande `ps2pdf`. Visualisez maintenant le résultat avec `xpdf`<sup>2</sup> ou ACROBAT READER, et découvrez les possibilités de navigation dans le document.

Vous pouvez obtenir le fichier PDF directement depuis le code source (`.tex`) avec la commande `pdflatex` (cela ne dispense pas de compiler deux fois).

### Premiers transparents

Un document pour BEAMER est avant tout structuré en sections et sous-sections. Les « cadres » sont ajoutés ensuite dans les différentes parties du document.

**4.** L'insertion d'images dans un document BEAMER se fait avec la commande vue au TD n° 1 : `\includegraphics`. Le format des images qu'il est possible d'insérer dépend de la commande utilisée pour compiler le document :

**latex** les images doivent être au format EPS ou PostScript ;

**pdflatex** les images doivent être au format PDF, JPEG ou PNG.

Insérez l'image `~ponsini/LaTeX/descartes2.jpg` dans le transparent intitulé *Motivation*.

**5.** Utilisez l'environnement `columns` pour placer l'image précédente à gauche du transparent

<sup>1</sup>Du fait des superpositions, un cadre peut être composé de plusieurs transparents ; par abus de langage nous employons parfois *transparent* pour désigner un *cadre*.

<sup>2</sup>L'option `-fullscreen` de `xpdf` permet d'afficher en mode plein écran.

alors que la partie droite contient une liste.

**6.** Dans le transparent de la section *Partie 1*, utilisez l'environnement `exampleblock` pour introduire un exemple. Cet environnement prend un argument qui sera le titre du bloc contenant l'exemple.

Beamer  
p. 99

## Superpositions

BEAMER propose un mécanisme pratique pour créer une animation comme par superposition de transparents, il comprend :

- un ensemble de commandes : `\onslide`, `\only`, `\uncover`, ...
- une notation pour ordonner les superpositions : `<intervalle, intervalle, ...>`

Beamer  
p. 61-73

**7.** Dans la section *Conclusion*, ajoutez un cadre contenant une liste dont les éléments apparaissent successivement.

**8.** Ajoutez un texte positionné avant la liste mais qui n'apparaîtra qu'après tous les éléments de la liste. Notez la différence entre `\onslide`, `\only` et `\visible`.

**9.** Ajoutez la commande `\setbeamercovered{transparent}` dans le préambule du document et essayez de nouveau les trois commandes de la question précédente.

Beamer  
p. 171

**10.** BEAMER dispose de deux commandes prédéfinies pour mettre en relief du texte par changement de couleur :

`\structure` la couleur utilisée est basée sur celle des éléments du décor du cadre ;

`\alert` la couleur utilisée est en général le rouge.

Ajoutez, à la suite de la liste de la section *Conclusion*, un texte qui change de couleur chaque fois qu'un élément de la liste apparaît.

Beamer  
p. 96-97

## Modèles prédéfinis

BEAMER fournit des modèles prédéfinis de présentation pour les différents éléments qui composent un transparent :

**theme** thème principal, affecte l'ensemble des éléments : `\usetheme`

**outer** affecte les éléments du décor des cadres : `\useoutertheme`

**inner** affecte les éléments qui ne sont pas du décor : `\useinnertheme`

**color** affecte les couleurs de la présentation : `\usecolortheme`

**font** affecte les polices de la présentation : `\usefonttheme`

Ces modèles se trouvent dans le répertoire :

`/usr/local/share/texmf-local/tex/latex/beamer/base/themes/`

Beamer  
p. 132

**11.** Essayez par exemple : `\usecolortheme{fly}` ou encore `\usetheme{Marburg}`.

## Plus de contrôle

En plus des thèmes prédéfinis, BEAMER offre aussi un contrôle manuel de tous les détails de la présentation. On utilise pour cela les commandes :

```
\setbeamertemplate{élément nommé}{définition}
\setbeamercolor{couleur nommée}{définition}
\setbeamerfont{police nommée}{définition}
```

Beamer  
p. 152,  
167 et  
177

**12.** Utilisez `\setbeamertemplate{navigation symbols}` et faites disparaître les symboles de navigation.

Beamer  
p. 55  
p. 48–50

**13.** Utilisez `\setbeamertemplate{footline}` et faites apparaître le numéro du transparent courant.

## Options de la classe beamer

**14.** Essayez les options suivantes :

**draft** permet de gagner du temps à la compilation en ne produisant pas les parties graphiques ;

**compress** réduit les barres de navigation ;

**notes=only** génère une version de la présentation avec les notes introduites par la commande `\note{contenu}` après le transparent concerné ;

**handout** génère une version imprimable. Pour grouper plusieurs transparents sur une page, utiliser `pdfnup` ou bien générer le `.ps` avec `dvips -ta4 in.dvi` puis utiliser `psnup` (par exemple : `psnup -4 -W128mm -H96mm -pa4 -b1 -m1 in.ps out.ps`).

Beamer  
p. 198

## Compléments pour vos documents

Les exercices de cette section sont à réaliser dans le document `rapport.tex` du TD n° 1.

### Bibliographie

Vous allez utiliser le programme `BIBTEX` pour inclure une bibliographie dans le rapport.

**15.** Copiez le fichier de bibliographie au format `BIBTEX ~ponsini/LaTeX/bibRapport.bib` et éditez-le avec EMACS. Ajoutez dans ce fichier, à l'aide des menus d'EMACS, les références d'un document de votre choix.

**16.** Remplacez la note en bas de page de la section 2.2 (*Listes et notes*) du rapport par des références aux deux documents sur Descartes. Citez ailleurs le document que vous avez ajouté.

Les citations se font avec la commande `\cite{clé}`. La clé est le premier élément après le type de l'entrée dans le fichier de bibliographie.

**17.** Pour insérer la bibliographie dans le document, normalement après les annexes, ajoutez les commandes suivantes :

```
\bibliographystyle{style}
\bibliography{nom du fichier de bibliographie sans le .bib}
```

Pour le style de la bibliographie, utilisez `plain` pour commencer.

**18.** Pour produire la bibliographie, compilez selon la séquence suivante :

1. `latex rapport.tex`
2. `bibtex rapport`
3. `latex rapport.tex` (deux fois)

p. 14  
et 73–75

## Code source

Pour les fragments de code source, il est d'usage d'utiliser une police de caractères différente du texte normal. De plus, pour être plus lisible, le code source est indenté.

**19.** L'environnement<sup>3</sup> `verbatim` peut servir pour la mise en forme de code source : dans cet environnement le texte apparaît tel qu'il a été saisi (espacement et retour à la ligne sont respectés et il n'y a plus de caractères spéciaux).

Dans l'annexe du rapport, ajoutez dans une figure un fragment de programme C en utilisant l'environnement `verbatim`.

**20.** Le paquetage `listings`<sup>4</sup> permet d'enjoliver les fragments de code source (numérotation des lignes, mots clés mis en relief...) pour un grand nombre de langages de programmation.

La commande `\lstset` permet de configurer la mise en forme du code source. Le code est ensuite inséré dans un environnement `\lstlisting` (pour les hors-texte) ou une commande `\lstinline` (pour un fragment dans une phrase de texte normal).

Toujours dans l'annexe, ajoutez la commande `\lstset{language=C}` puis utilisez l'environnement `lstlisting` pour mettre en forme le même fragment de programme que dans l'exercice précédent.

---

<sup>3</sup>Pour un fragment de code dans une phrase de texte normal, on pourra utiliser la commande `\verb` (cf. *L<sup>A</sup>T<sub>E</sub>X* en bref p. 6–7).

<sup>4</sup>La documentation se trouve sur uranie dans `/usr/local/share/texmf/doc/latex/listings/`