

Organisation de l'enseignement

- 1^{er} cours : présentation et utilisation élémentaire de \LaTeX
- 2^{ème} cours : mode mathématique et utilisation avancée de \LaTeX
 - formules mathématiques
 - présentations avec la classe Beamer
 - \LaTeX (un peu) avancé
- 3^{ème} cours : utilisation avancée (suite) et conseils pour le TE

Mode mathématique

- Mode spécifique, avec conventions différentes des autres :
 - les espaces sont entièrement ignorés
 - les variables ne font normalement qu'une seule lettre
 - \wedge et $_$ servent de commandes
 - pas de passage à la ligne dans une formule
- Plusieurs manières d'entrer dans le mode :
 - $\$$ et $\$,$ mais avec possibilité d'erreur
 - \langle et $\rangle,$ bien parenthésé mais *fragile*
 - $\$\$$ et $\$\$,$ ou \langle et $\rangle,$ pour les formules hors texte
 - déclaration `\ensuremath` pour passer en mode mathématique dans une commande

Règles de base

- Une formule dans le texte et une formule isolée ne sont pas présentées de la même manière.
- Certains environnements passent en mode mathématique (par exemple `equation`).
- D'autres environnements nécessitent qu'on y soit (par exemple `array`).
- Normalement, la maîtrise de l'espacement est entièrement déléguée à \LaTeX .

Environnements spécifiques

- `array` est comme `tabular`, mais tout entier en mode mathématique.
- `equation` fournit une formule isolée, centrée, et numérotée entre parenthèses et à droite.
- `eqnarray` permet de placer une suite de formules ou d'équations ou de formules numérotées :
 - tableau sur trois colonnes
 - en mode mathématique
 - chaque ligne est numérotée sauf si apparaît `\nonumber` en fin de ligne

Opérateurs

- Opérateurs binaires :
 - on trouve les opérateurs habituels : + - /
 - des opérateurs codés : `\div \times \cap \cup` ÷ × ∩ ∪
 - les opérateurs de comparaison : `<=>`
 - mais aussi codés : `\leq \subset \neq` ≤ ⊂ ≠
 - on peut nier un opérateur existant : `\not\leq` ≰
- Points d'énumération :
 - sur la ligne : `\ldots` ...
 - au niveau de la barre horizontale de + : `\cdots` ...
 - verticaux : `\vdots` ∴
 - descendants : `\ddots` ∴

Opérateurs composés

- puissance `\(X^{a-b}\)` X^{a-b}
- indice `\mathbb{T}_{a_i}` T_{a_i}
- les deux ensemble :
 - `\sum_{i=1}^n` $\sum_{i=1}^n$
 - `\[\sum_{i=1}^n\]`

$$\sum_{i=1}^n$$

Opérateurs à deux dimensions

- fraction `\frac{numérateur}{dénominateur}`
- `\frac{ax^2+bx+c}{a-bx}` $\frac{ax^2+bx+c}{a-bx}$
 - `\[\frac{ax^2+bx+c}{a-bx}\]`
- $$\frac{ax^2 + bx + c}{a - bx}$$
- racine `\sqrt{expression}` ou `\sqrt[nième]{expr}`
- `\(\sqrt{ax^2+bx+c}\)` $\sqrt{ax^2 + bx + c}$
 - `\[\sqrt[n+2]{ax^2+bx+c}\]`
- $$\sqrt[n+2]{ax^2 + bx + c}$$

Symboles de taille variable Opérateurs

- `\sum \prod \int \bigcup`

$$\Sigma \Pi / U$$

- tailles différentes dans le texte ou en évidence : $\sum_0^n i$
- indices et exposants placés d'après le symbole :

$$\int_{-\infty}^{+\infty} \sum_0^n e^x e^y$$

Symboles de taille variable

Délimiteurs

- en particulier toutes les parenthèses
- `\left` ou `\right` précèdent le délimiteur
- la symétrie n'est pas nécessaire, mais il faut fermer le parenthésage, au moins avec le `point` qui sert de délimiteur invisible

Exemple

```

\[\left(\begin{array}{c}
\left(\begin{array}{cc}
x_{11}& x_{12} \\
x_{21}& x_{22} \\
\vdots
\end{array}\right)
\end{array}\right)\]

```

$$\left(\left(\begin{array}{cc} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \end{array} \right) \right)$$

Autres possibilités

- Traits et flèches :
 - `\overrightarrow` $\overrightarrow{ax^2 - b}$
 - `\Rightarrow` et `\stackrel{G}{\rightarrow}` $A \stackrel{G}{\rightarrow} x - y$
- Accents :
 - `\vec` \vec{n}
 - `\breve` \breve{z}
 - `\widehat` $\widehat{1-x}$
- Polices :
 - `\mathcal` \mathcal{AMS}
 - `\mathbb` \mathbb{NZRC}
 - `\mathfrak` $\mathfrak{E} = mc^2$
- Symboles :
 - `\forall` `\in` `\Sigma` $\forall i \in \Sigma$
 - `\Re` `\nabla` `\aleph` $\Re \nabla \aleph$
- Très nombreuses extensions de l'AMS

La classe de document Beamer

- Présentations pour vidéoprojecteur ou rétroprojecteur
- Fidèle à l'esprit de LaTeX
- Nombreux thèmes prédéfinis, récents et réellement utiles
- Possibilités d'animation et de lien hypertexte
- Gestion des différentes versions d'une présentation
- Bonne documentation

Structure d'un document Beamer

- Préambule
- Sections et sous-sections
- Cadres
- Superpositions
- Environnements structurants standards et spécifiques

Structure d'un document Beamer

Préambule

Le préambule permet de composer un style de présentation global à l'aide de thèmes prédéfinis facultatifs :

- thème principal : `\usetheme[options]{nom}`
- variations autour d'un thème :
 - couleurs : `\usecolortheme[options]{nom}`
 - polices : `\usefonttheme[options]{nom}`
 - éléments d'un cadre : `\useinnertheme[options]{nom}`
 - éléments hors cadre : `\useoutertheme[options]{nom}`

D'autres mécanismes permettent de contrôler dans le détail l'apparence de n'importe quel élément.

Les « cadres »

(frames)

Les cadres sont créés et contenus par l'environnement `frame` :

- `\begin{frame}[options] ... \end{frame}`
- parmi les options :
 - `b`, `t` ou `c` position des éléments dans le cadre
 - `plain` suppression des éléments autour du cadre
- titres : `\frametitle{titre}` et `\framesubtitle{titre}`

Les « cadres »

Composants

Les éléments constitutifs d'un cadre sont :

- un en-tête et un pied de page
- une marge gauche et une marge droite
- des barres de navigation
- des symboles de navigation
- un logo
- un titre de cadre
- un fond
- un contenu

Les « cadres »

Superpositions (overlays)

- Les cadres sont constitués d'un ou plusieurs « transparents » qui peuvent être superposés.
- Les transparents nécessaires sont automatiquement calculés et numérotés.
- Pour spécifier sur quels transparents du cadre une commande doit être appliquée, on la fait suivre de la notation `<plage>` :
 - `<1>` désigne l'unique transparent où apparaîtra l'élément
 - `<1,3,7>` désigne une liste de transparents
 - `<2-5>` désigne un intervalle, bornes incluses
 - `<-2>` `<4->` désigne un intervalle ouvert (la borne manquante est remplacée par le numéro du premier ou du dernier transparent du cadre)
 - `<-3,5,7->` on peut combiner les écritures

Les « cadres »

Commandes et superpositions

Les commandes spécifiques suivantes acceptent une spécification de superposition :

- `\visible<s>\{texte}` : le texte en argument n'est visible que sur les transparents spécifiés
- `\only<s>\{texte}` : *idem*, mais n'occupe aucun espace sur les autres transparents
- `\uncover<s>\{texte}` : *idem* « visible » mais, selon le thème, le texte est en filigrane sur les transparents non spécifiés
- `\onslide<s>` : *idem* mais pour tout le texte qui suit dans le cadre courant

Les « cadres »

Commandes et superpositions

Quelques autres commandes qui acceptent une spécification de superposition :

- `\item<s>`
- les commandes sur les polices, ex : `\textbf<s>`
- `\alert<s>\{texte}`
- la plupart des commandes et environnements propres à Beamer

Les « cadres »

Commandes et superpositions

Exemple

Un `\alert<4>\{exemple}`

`\only<2>\{(insertion du texte)}` :

`\begin{itemize}`

`\item<1-> elt 1`

`\item<2-> elt 2`

`\item<3-> elt 3`

`\end{itemize}`

Un **exemple** (insertion du texte) :

- elt 1
- elt 2
- elt 3

Environnements spécifiques

Quelques environnements additionnels de Beamer :

- `block` un bloc de texte avec un titre passé en argument
- `exampleblock` pour les exemples, le titre est passé en argument
- `columns` partage le cadre en colonnes
- `column` dans l'environnement `columns`, crée une colonne dont la taille est passée en argument

Environnements spécifiques

Exemple

```
\begin{columns}
```

```
\begin{column}{0.5\textwidth}
```

Le premier étoit de ne recevoir jamais aucune chose pour vraie que je ne la connusse évidemment être telle ;

```
\end{column}
```

```
\begin{column}{0.5\textwidth}
```

Le second, de diviser chacune des difficultés que j'examinerois, en autant de parcelles qu'il se pourroit, et qu'il seroit requis pour les mieux résoudre.

```
\end{column}
```

```
\end{columns}
```

Programmation de L^AT_EX

- La programmation de L^AT_EX utilise à la fois les outils de bas niveau de T_EX et des mécanismes de plus haut niveau introduits par L^AT_EX.
- Le mélange des deux types n'est pas souhaitable (lisibilité, compatibilité).
- Nous ne verrons que les mécanismes propres à L^AT_EX.
- Les objets de base sont les suivants :
 - les compteurs
 - les longueurs
 - les commandes pourront servir de variables à valeur de chaîne de caractères

Compteurs

- Valeur numérique entière
- Déclaration par `\newcounter{compteur}`
- Affectation par `\setcounter{compteur}{valeur}`
- Augmentation par `\addtocounter{compteur}{valeur}`
- Utilisation de la valeur comme un entier par `\value{compteur}`
- Utilisation de la valeur comme une chaîne par `\the compteur`
- Le nom de compteur est un symbole quelconque, généralement un identificateur.

Compteurs

Exemple

```
\newcounter{toto}
```

```
\setcounter{toto}{2}
```

La valeur du compteur `\emph{toto}` est `\thetoto`

```
\addtocounter{toto}{\value{toto}}
```

et maintenant `\thetoto`.

La valeur du compteur *toto* est 2 et maintenant 4.

Longueurs

- Valeur formée de trois composants :
 - longueur proprement dite : nombre décimal suivi de l'unité de longueur
 - élasticité positive
 - élasticité négative
- S'il n'y a pas d'élasticité, la longueur est **rigide**.
- Une longueur élastique sert pour la colle entre deux boîtes, horizontalement ou verticalement.
- `\fill` est une longueur infiniment élastique.

Longueurs

Unités principales

- unités absolues : **cm**, **mm**
 - in** = 2,54 cm
 - pt** = 0,351 mm (point pica)
 - bp** = 0,3527 mm (point Postscript)
 - dd** = 0,376 mm (point Didot)
 - sp** = $5,356 \times 10^{-6}$ mm (point T_EX)
- unités relatives :
 - ex** = hauteur de la lettre **x** dans la police en cours
 - em** = largeur de la lettre **M** dans la police en cours

Longueurs

Commandes

- Déclaration par `\newlength{longueur}`
- Affectation par `\setlength{longueur}{valeur}`
- Augmentation par `\addtolength{longueur}{valeur}`
- Affectation de la largeur d'un texte par `\settowidth{longueur}{texte}`
- Affectation de la hauteur d'un texte par `\settoheight{longueur}{texte}`
- le nom de longueur est un nom de commande

Exemple

```
\newlength{\largepage}
\setlength{\largepage}{\textwidth}
\addtolength{\largepage}{2.5cm}
\newlength{\tailletoto}
\settowidth{\tailletoto}{\Large TOTO}
```

Commandes et environnements

Règles d'écriture

- Un nom de **commande** commence par `\` et est une suite de lettres, sauf les cas rares où c'est un opérateur (diacritiques par exemple).
- Un nom d'**environnement** est une suite de lettres.
- Majuscules et minuscules sont significatives
- Chiffres et signes de ponctuation sont interdits dans les noms.

Commandes et environnements

Arguments

Une commande ou un environnement peuvent avoir des arguments :

- Leur nombre va de **zéro à neuf** arguments en tout.
- Le **premier** peut être facultatif, il est spécifié entre crochets s'il apparaît.
- Les autres sont spécifiés entre accolades.
- Attention aux blancs dans les arguments : ils comptent pour un blanc.
- Des blancs entre les arguments sont ignorés.

Attention :

`\item { [pas une étiquette] }` • `[pas une étiquette]`

Commandes et environnements

Déclarations de commandes

- Trois déclarations de même forme :
 - `\newcommand` pour créer une commande qui n'existait pas
 - `\renewcommand` pour redéfinir une commande existante
 - `\providecommand` pour créer une commande si elle n'existait pas déjà
- Forme générale :
 - `\newcommand{\nom}[nombre][défaut]{contenu}`

Commandes et environnements

`\newcommand{\nom}[nombre][défaut]{contenu}`

- *nombre* est le nombre d'arguments, ne rien mettre s'il n'y en a pas.
- *défaut* est la valeur par défaut de l'argument facultatif (ce qui indique qu'il y en a un).
- Dans le *contenu*, la référence aux arguments est indiquée par `#n`, où *n* est un chiffre de 1 à 9
- Les accolades sont supprimées au moment de la substitution.

Commandes et environnements

Exemples

- Simple abréviation :


```
\newcommand{\flr}{\ensuremath{\Longrightarrow}}
```
- Commande avec argument facultatif :


```
\newcommand{\PK}[1][0]{\ensuremath{\mathrm{K}^{\#1}}}
```
- Commande avec plusieurs arguments :


```
\newcommand{\classement}[3]%
{\begin{enumerate}%
\item #1%
\item #2%
\item #3%
\end{enumerate}}
```


Commandes et environnements

Déclarations d'environnements

- Déclarations similaires aux commandes : `\newenvironment`, `\renewenvironment`, `\provideenvironment`
- Forme générale :
`\newenvironment{nom}[nombre][défaut]%
{début}{fin}`
- Règles :
 - Définit à la fois `\begin{nom}` et `\end{nom}`
 - Seul `\begin{nom}` peut avoir des arguments.
 - Les références aux arguments ne peuvent apparaître que dans *début*.
 - `\begin{nom}` est équivalent à `\nom` et `\end{nom}` à `\endnom`

Commandes et environnements

Exemples

- Environnement sans argument :
`\newenvironment{exemple}%
{\begin{quote}{\slshape Exemple} :\\}%
{\(\diamond)\end{quote}}`
- Environnement avec un argument :
`\newenvironment{reglenom}[1]%
{\begin{trivlist}
\item[(#1)]}%
{\end{trivlist}}`

Alignement

Un paragraphe normal est **justifié** :

Le bon sens est la chose du monde la mieux partagée; car chacun pense en être si bien pourvu que ceux même qui sont les plus difficiles à Contenter en toute autre chose n'ont point coutume d'en désirer plus qu'ils en ont.

Le bon sens est la chose du monde la mieux partagée; car chacun pense en être si bien pourvu que ceux même qui sont les plus difficiles à Contenter en toute autre chose n'ont point coutume d'en désirer plus qu'ils en ont.

Alignement

On peut changer temporairement la présentation des paragraphes par plusieurs environnements :

- `flushright` pour aligner les lignes sur la droite
- `flushleft` pour les aligner sur la gauche (composition « en drapeau »)
- `center` pour les centrer
- `quote` pour ajouter un retrait à gauche et à droite
- `quotation` pour faire la même chose pour plusieurs alinéas successifs
- `verse` pour de la poésie, les vers étant séparés par la commande de fin de ligne : `\newline` ou `\\`

Alignement

Exemple

```
\begin{flushright}
La citation suivante est du même auteur :
\begin{quote}
\og je pense, donc je suis \fg
\end{quote}
\end{flushright}
```

La citation suivante est du même auteur :
« *je pense, donc je suis* »

Polices
Changement de corps

<code>\tiny tiny</code>	tiny
<code>\scriptsize script</code>	script
<code>\footnotesize foot</code>	foot
<code>\small small</code>	small
<code>\normalsize normal</code>	normal
<code>\large large</code>	large
<code>\Large Large</code>	Large
<code>\LARGE LARGE</code>	LARGE
<code>\huge huge</code>	huge
<code>\Huge Huge</code>	Huge

Polices
Changement de polices

- Par déclaration :
 - famille :

<code>\rmfamily romain</code>	romain
<code>\sffamily sans</code>	sans empattements
<code>\ttfamily tty</code>	chasse fixe
 - série :

<code>\bfseries gras</code>	gras
<code>\mdseries médium</code>	médium
 - forme :

<code>\itshape italique</code>	<i>italique</i>
<code>\upshape droit</code>	droit
<code>\slshape penché</code>	<i>penché</i>
<code>\scshape petcap</code>	PETITES CAPITALES

Polices
Changement de polices

- Par commande :

<code>\textrm{}</code>	romain
<code>\textit{}</code>	<i>italique</i>
<code>\textsf{}</code>	sans empattements
<code>\textsl{}</code>	<i>penché</i>
<code>\textbf{}</code>	gras
<code>\textsc{}</code>	PETITES CAPITALES
<code>\texttt{}</code>	chasse fixe

Pour changer les polices disponibles par défaut :

- utiliser des paquetages spécifiques
- utiliser la déclaration de bas niveau
`\usefont{codage}{nom}{série}{forme}`