

Introduction à L^AT_EX

Introduction à l'utilisation des bases de données mathématiques
Méthodologie de la recherche
Formation au L^AT_EX

1^{ère} partie

Karl GROSSE-ERDMANN Aline GOULARD



16 février 2017

① Introduction

Introduction

\LaTeX , ça sert à quoi ?

\LaTeX est un logiciel gratuit pour

- écrire des textes scientifiques (contenant des symboles mathématiques)
- faire des présentations (comme celle-ci)
- plein d'autres choses : écrire n'importe quel texte, écrire des lettres, faire des tableaux, faire des graphiques,...

Introduction

\LaTeX , ça sert à quoi ?

\LaTeX est un logiciel gratuit pour

- écrire des textes scientifiques (contenant des symboles mathématiques)
- faire des présentations (comme celle-ci)
- plein d'autres choses : écrire n'importe quel texte, écrire des lettres, faire des tableaux, faire des graphiques,...

Introduction

\LaTeX , ça sert à quoi ?

\LaTeX est un logiciel gratuit pour

- écrire des textes scientifiques (contenant des symboles mathématiques)
- faire des présentations (comme celle-ci)

• plein d'autres choses : écrire n'importe quel texte, écrire des lettres, faire des tableaux, faire des graphiques, ...

Introduction

\LaTeX , ça sert à quoi ?

\LaTeX est un logiciel gratuit pour

- écrire des textes scientifiques (contenant des symboles mathématiques)
- faire des présentations (comme celle-ci)
- plein d'autres choses : écrire n'importe quel texte, écrire des lettres, faire des tableaux, faire des graphiques,...

Un peu d'histoire

1978 Donald E. Knuth, un informaticien, crée $\text{T}_{\text{E}}\text{X}$

1983-85 L'American Mathematical Society crée $\mathcal{A}_{\text{M}}\text{S-}\text{T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1985 Leslie Lamport crée \LaTeX , basé sur $\text{T}_{\text{E}}\text{X}$

1990 L'AMS crée $\mathcal{A}_{\text{M}}\text{S-}\text{\LaTeX}$

1994 $\text{\LaTeX} 2_{\epsilon}$ est créé

1994– travail sur une version $\text{\LaTeX} 3$ (le projet $\text{\LaTeX} 3$)

Un peu d'histoire

1978 Donald E. Knuth, un informaticien, crée $\text{T}_{\text{E}}\text{X}$

1983-85 L'American Mathematical Society crée $\text{A}_{\text{M}}\text{S-}\text{T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1985 Leslie Lamport crée $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1990 L'AMS crée $\text{A}_{\text{M}}\text{S-}\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$

1994 $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ est créé

1994– travail sur une version $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 3$ (le projet $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X} 3$)

Un peu d'histoire

1978 Donald E. Knuth, un informaticien, crée $\text{T}_{\text{E}}\text{X}$

1983-85 L'American Mathematical Society crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1985 Leslie Lamport crée \LaTeX , basé sur $\text{T}_{\text{E}}\text{X}$

1989 L'AMS crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-}\text{\LaTeX}$

1994 $\text{\LaTeX} 2_{\epsilon}$ est créé

1994- travail sur une version $\text{\LaTeX} 3$ (le projet $\text{\LaTeX} 3$)

Un peu d'histoire

1978 Donald E. Knuth, un informaticien, crée $\text{T}_{\text{E}}\text{X}$

1983-85 L'American Mathematical Society crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1985 Leslie Lamport crée \LaTeX , basé sur $\text{T}_{\text{E}}\text{X}$

1987 L'AMS crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-}\text{\LaTeX}$

1992 $\text{\LaTeX} 2_{\epsilon}$ est créé

1994- travail sur une version $\text{\LaTeX} 3$ (le projet $\text{\LaTeX} 3$)

Un peu d'histoire

1978 Donald E. Knuth, un informaticien, crée $\text{T}_{\text{E}}\text{X}$

1983-85 L'American Mathematical Society crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1985 Leslie Lamport crée $\text{L}\text{A}\text{T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1990 L'AMS crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-L}\text{A}\text{T}_{\text{E}}\text{X}$

$\text{L}\text{A}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$ est créé

1996 – travail sur une version $\text{L}\text{A}\text{T}_{\text{E}}\text{X} 3$ (le projet $\text{L}\text{A}\text{T}_{\text{E}}\text{X} 3$)

Un peu d'histoire

1978 Donald E. Knuth, un informaticien, crée $\text{T}_{\text{E}}\text{X}$

1983-85 L'American Mathematical Society crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1985 Leslie Lamport crée \LaTeX , basé sur $\text{T}_{\text{E}}\text{X}$

1990 L'AMS crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-}\text{\LaTeX}$

1994 $\text{\LaTeX} 2_{\epsilon}$ est créé

2006 – travail sur une version $\text{\LaTeX} 3$ (le projet $\text{\LaTeX} 3$)

Un peu d'histoire

1978 Donald E. Knuth, un informaticien, crée $\text{T}_{\text{E}}\text{X}$

1983-85 L'American Mathematical Society crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1985 Leslie Lamport crée \LaTeX , basé sur $\text{T}_{\text{E}}\text{X}$

1990 L'AMS crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-}\text{\LaTeX}$

1994 $\text{\LaTeX} 2_{\epsilon}$ est créé

1994– travail sur une version $\text{\LaTeX} 3$ (le projet $\text{\LaTeX} 3$)

Normalement, on utilise $\text{\LaTeX} 2_{\epsilon}$ en y ajoutant des macros (packages).

Un peu d'histoire

1978 Donald E. Knuth, un informaticien, crée $\text{T}_{\text{E}}\text{X}$

1983-85 L'American Mathematical Society crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-T}_{\text{E}}\text{X}$, basé sur $\text{T}_{\text{E}}\text{X}$

1985 Leslie Lamport crée \LaTeX , basé sur $\text{T}_{\text{E}}\text{X}$

1990 L'AMS crée $\mathcal{A}_{\mathcal{M}}\mathcal{S}\text{-}\text{\LaTeX}$

1994 $\text{\LaTeX} 2_{\epsilon}$ est créé

1994– travail sur une version $\text{\LaTeX} 3$ (le projet $\text{\LaTeX} 3$)

Normalement, on utilise $\text{\LaTeX} 2_{\epsilon}$ en y ajoutant des macros (packages).

Trois exemples

L^AT_EX de base :

$$\lim_{n \rightarrow +\infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

avec le package musixtex :

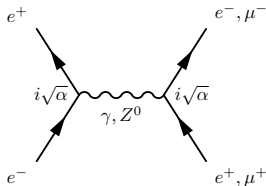


Trois exemples

L^AT_EX de base :

$$\lim_{n \rightarrow +\infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

avec le package `feynmp` :



avec le package `musixtex` :

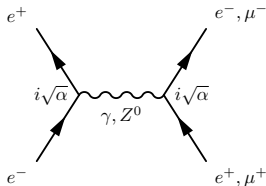


Trois exemples

L^AT_EX de base :

$$\lim_{n \rightarrow +\infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

avec le package `feynmp` :



avec le package `musictex` :



Comment ça marche ?

1. Ecrire un document \LaTeX (extension `.tex`) en utilisant un éditeur
2. Compiler ce document pour créer un fichier qui peut être visualisé et imprimé (extensions : principalement `.pdf`)



NB. La compilation crée d'autres fichiers automatiquement (un fichier `.log` avec des informations sur la compilation, un fichier `.aux` avec des informations sur les références, etc.).

Comment ça marche ?

1. Ecrire un document \LaTeX (extension `.tex`) en utilisant un éditeur
2. Compiler ce document pour créer un fichier qui peut être visualisé et imprimé (extensions : principalement `.pdf`)



NB. La compilation crée d'autres fichiers automatiquement (un fichier `.log` avec des informations sur la compilation, un fichier `.aux` avec des informations sur les références, etc.).

Comment ça marche ?

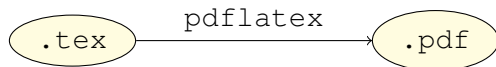
1. Ecrire un document \LaTeX (extension `.tex`) en utilisant un éditeur
2. Compiler ce document pour créer un fichier qui peut être visualisé et imprimé (extensions : principalement `.pdf`)



NB. La compilation crée d'autres fichiers automatiquement (un fichier `.log` avec des informations sur la compilation, un fichier `.aux` avec des informations sur les références, etc.).

Comment ça marche ?

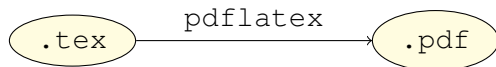
1. Ecrire un document \LaTeX (extension `.tex`) en utilisant un éditeur
2. Compiler ce document pour créer un fichier qui peut être visualisé et imprimé (extensions : principalement `.pdf`)



NB: La compilation crée d'autres fichiers automatiquement (un fichier `.log` avec des informations sur la compilation, un fichier `.aux` avec des informations sur les références, etc.).

Comment ça marche ?

1. Ecrire un document \LaTeX (extension `.tex`) en utilisant un éditeur
2. Compiler ce document pour créer un fichier qui peut être visualisé et imprimé (extensions : principalement `.pdf`)



NB. La compilation crée d'autres fichiers automatiquement (un fichier `.log` avec des informations sur la compilation, un fichier `.aux` avec des informations sur les références, etc.).

Que faut-il ?

- une distribution (ensemble de packages fréquemment utilisés),

Que faut-il ?




- ❶ une distribution (ensemble de packages fréquemment utilisés),
- ❷ un éditeur (logiciel qui permet d'écrire et de modifier des fichiers texte et qui en général offre de l'aide pour respecter la syntaxe spécifique requise par le document).

Que faut-il ?

- 1 une distribution (ensemble de packages fréquemment utilisés),
- 2 un éditeur (logiciel qui permet d'écrire et de modifier des fichiers texte et qui en général offre de l'aide pour respecter la syntaxe spécifique requise par le document).

Les distributions et les éditeurs

Les plus courants

OS	Distribution	Exemple d'éditeurs
	MiKTeX	Texmaker, TeXnicCenter
	MacTeX	TeXShop, iTeXMac
	TeXLive	Emacs (avec le package AUCTEX)

Certains éditeurs et distributions existent sur plusieurs systèmes d'exploitations. On ne cite ici que les plus fréquents pour chacun des systèmes d'exploitation.

Comment les obtenir ?

Exemple : MikTeX et Texmaker sous Windows

Comment les obtenir ?

Exemple : MiKTeX et Texmaker sous Windows

Comment les obtenir ?

Exemple : MiKTeX et Texmaker sous Windows

Télécharger et installer le 'basic MiKTeX system' :

<http://miktex.org/download>

Télécharger et installer Texmaker :

http://www.xmlmath.net/texmaker/index_fr.html

Comment les obtenir ?

Exemple : MiKTeX et Texmaker sous Windows

Télécharger et installer le 'basic MiKTeX system' :

<http://miktex.org/download>

Télécharger et installer Texmaker :

http://www.xmlmath.net/texmaker/index_fr.html

Informations sur L^AT_EX

- Nicolas Markey : Introduction à L^AT_EX (2002)
<http://www.lsv.ens-cachan.fr/~markey/bibla.php>
- George Grätzer : More math into L^AT_EX, Springer 2007
<http://link.springer.com/book/10.1007/978-0-387-68852-7>
- George Grätzer : Practical L^AT_EX, Springer 2014
<http://link.springer.com/book/10.1007/978-3-319-06425-3>
- GUTenberg, le groupe francophone des utilisateurs de T_EX,
<http://www.gutenberg.eu.org/>

Informations sur L^AT_EX

- Nicolas Markey : Introduction à L^AT_EX (2002)
<http://www.lsv.ens-cachan.fr/~markey/bibla.php>
- George Grätzer : More math into L^AT_EX, Springer 2007
<http://link.springer.com/book/10.1007/2F978-0-387-68852-7>
- George Grätzer : Practical L^AT_EX, Springer 2014
<http://link.springer.com/book/10.1007/978-3-319-06425-3>
- GUTenberg, le groupe francophone des utilisateurs de T_EX,
<http://www.gutenberg.eu.org/>

Le document .tex

Introduction à LaTeX

La grande majorité des commandes en \LaTeX commence par le caractère `\`, leurs arguments obligatoires sont insérés dans des accolades et leurs arguments optionnels entre crochets.

Exemples :

`\alpha` α

`\sqrt{2}` $\sqrt{2}$

`\sqrt[3]{2}` $\sqrt[3]{2}$

Principe fondamental

La grande majorité des commandes en \LaTeX commence par le caractère `\`, leurs arguments obligatoires sont insérés dans des accolades et leurs arguments optionnels entre crochets.

Exemples

`\alpha` α

`\sqrt{2}` $\sqrt{2}$

`\sqrt[3]{2}` $\sqrt[3]{2}$

Principe fondamental

La grande majorité des commandes en \LaTeX commence par le caractère `\`, leurs arguments obligatoires sont insérés dans des accolades et leurs arguments optionnels entre crochets.

Exemples :

`\alpha` α

`\sqrt{2}` $\sqrt{2}$

`\sqrt[3]{2}` $\sqrt[3]{2}$

Principe fondamental

La grande majorité des commandes en \LaTeX commence par le caractère `\`, leurs arguments obligatoires sont insérés dans des accolades et leurs arguments optionnels entre crochets.

Exemples :

`\alpha` α

`\sqrt{2}` $\sqrt{2}$

`\sqrt[3]{2}` $\sqrt[3]{2}$

Principe fondamental

La grande majorité des commandes en \LaTeX commence par le caractère `\`, leurs arguments obligatoires sont insérés dans des accolades et leurs arguments optionnels entre crochets.

Exemples :

`\alpha` α

`\sqrt{2}` $\sqrt{2}$

`\sqrt[3]{2}` $\sqrt[3]{2}$

Principe fondamental

La grande majorité des commandes en \LaTeX commence par le caractère `\`, leurs arguments obligatoires sont insérés dans des accolades et leurs arguments optionnels entre crochets.

Exemples :

`\alpha` α

`\sqrt{2}` $\sqrt{2}$

`\sqrt[3]{2}` $\sqrt[3]{2}$

② Documents simples

Structure générale (1/4)

```
\documentclass[a]{b}
```

préambule

```
\begin{document}
```

texte

```
\end{document}
```

a sont les options du document séparées par des virgules :
la taille de la police (ex : 12pt), le format papier (ex :
a4paper), ...

b est le « type » (classe) du document :
article, report, book ou également beamer, ...

Structure générale (1/4)

```
\documentclass[a]{b}
```

préambule

```
\begin{document}
```

texte

```
\end{document}
```

- a** sont les options du document séparées par des virgules :
la taille de la police (ex : 12pt), le format papier (ex : a4paper), ...
- b** est le « type » (classe) du document :
article, report, book ou également beamer, ...

Structure générale (2/4)

- `article` généralement utilisée pour les articles scientifiques
- `report` identique à `article` mais convient mieux aux mémoires, petits livres, thèses
- `book` convient pour l'édition des livres et les documents volumineux
- `beamer` pour les transparents

Structure générale (3/4)

Le préambule spécifie, par exemple, les packages qu'il faudra charger afin d'utiliser certaines commandes.

Certains packages sont vivement conseillés dans tous les cas :

`\usepackage{utf8}` (pour `utf8`)
(et bien vérifier que son éditeur encode en UTF-8)

`\usepackage[francais]{babel}` ou
`\usepackage[francais]{babel}`
permettent de définir l'affichage des mots comme « chapitre »,
« section » et la typographie en français.

`\usepackage{amsmath}`, `amstext`, `amsymb`, `amsthm`,
pour l'écriture des symboles mathématiques.

Structure générale (3/4)

Le préambule spécifie, par exemple, les packages qu'il faudra charger afin d'utiliser certaines commandes.

Certains packages sont vivement conseillés dans tous les cas :

`\usepackage[utf8]{inputenc}`
(et bien vérifier que son éditeur encode en UTF-8)

`\usepackage[french]{babel}` ou
`\usepackage[french]{babel}`
permettent de définir l'affichage des mots comme « chapitre »,
« section » et la typographie en français.

`\usepackage{amsmath}` et `\usepackage{amssymb}` sont
pour l'écriture des symboles mathématiques.

Structure générale (3/4)

Le préambule spécifie, par exemple, les packages qu'il faudra charger afin d'utiliser certaines commandes.

Certains packages sont vivement conseillés dans tous les cas :

```
\usepackage[utf8]{inputenc}
```

(et bien vérifier que son éditeur encode en UTF-8)

```
\usepackage[frenchb]{babel} ou
```

```
\usepackage[français]{babel}
```

permettent de définir l'affichage des mots comme « chapitre », « section » et la typographie en français.

```
\usepackage{amsmath,amsfonts,amssymb,amsthm}
```

pour l'écriture des symboles mathématiques.

Structure générale (4/4)

Quelques remarques sur la structure du texte :

Structure générale (4/4)

Quelques remarques sur la structure du texte :

- Une ligne vide crée un nouveau paragraphe.
 - La commande `\n` signifie un retour à la ligne sans qu'un nouveau paragraphe ne soit créé (c'est déconseillé).
 - Avec `\noindent` on peut localement supprimer l'alinéa d'un nouveau paragraphe.

Structure générale (4/4)

Quelques remarques sur la structure du texte :

- Une ligne vide crée un nouveau paragraphe.
- La commande `\\` signifie un retour à la ligne sans qu'un nouveau paragraphe ne soit créé (c'est déconseillé).

• Avec `\section{}` on peut localement supprimer l'alinéa d'un nouveau paragraphe.

Structure générale (4/4)

Quelques remarques sur la structure du texte :

- Une ligne vide crée un nouveau paragraphe.
- La commande `\` signifie un retour à la ligne sans qu'un nouveau paragraphe ne soit créé (c'est déconseillé).
- Avec `\noindent` on peut localement supprimer l'alinéa d'un nouveau paragraphe.

Les modes mathématiques (1/4)

Toute expression mathématique doit être incorporée dans un mode mathématique.

Le mode simple : $\$...\$$ au milieu du texte

Exemple :

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

donnera

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

Les modes mathématiques (1/4)

Toute expression mathématique doit être incorporée dans un mode mathématique.

Le mode simple : $\$ \dots \$$ au milieu du texte

Exemple :

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

donnera

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

Les modes mathématiques (1/4)

Toute expression mathématique doit être incorporée dans un mode mathématique.

Le mode simple : $\$ \dots \$$ au milieu du texte

Exemple :

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

donnera

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

Les modes mathématiques (1/4)

Toute expression mathématique doit être incorporée dans un mode mathématique.

Le mode simple : $\$ \dots \$$ au milieu du texte

Exemple :

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

donnera

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

Les modes mathématiques (2/4)

Le mode mathématique centré (passer à la ligne et centrer) :
peut être obtenu par

`\[`

`...`

`\]`

Il est déconseillé d'utiliser la commande équivalente

`$$...$$`

Une autre alternative est

`\begin{equation} ... \end{equation}`

Si on veut numéroté l'équation, on utilise

`\begin{equation} ... \end{equation}`

Les modes mathématiques (2/4)

Le mode mathématique centré (passer à la ligne et centrer) : peut être obtenu par

```
\[  
...  
\]
```

Il est déconseillé d'utiliser la commande équivalente

```
$$...$$
```

Une autre alternative est

```
\begin{equation*}...\end{equation*}
```

Si on veut numéroté l'équation, on utilise

```
\begin{equation}...\end{equation}
```

Les modes mathématiques (3/4)

Exemple :

prenons en particulier

```
\[  
y=\sum_{n=1}^5 a_n  
\]
```

dans l'expression précédente...

donnera

prenons en particulier

$$y = \sum_{n=1}^5 a_n$$

dans l'expression précédente...

Les modes mathématiques (3/4)

Exemple :

prenons en particulier

```
\[  
y=\sum_{n=1}^5 a_n  
\]
```

dans l'expression précédente...

donnera

prenons en particulier

$$y = \sum_{n=1}^5 a_n$$

dans l'expression précédente...

Les modes mathématiques (4/4)

Avec `\displaystyle` dans le mode simple, on peut forcer \LaTeX à afficher les mathématiques comme si elles étaient dans le mode mathématique centré.

Exemple :

prenons en particulier `\displaystyle y=\sum_{n=1}^5 a_n` dans l'expression précédente...

donnera

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

La commande `\textstyle` à l'effet inverse.

Les modes mathématiques (4/4)

Avec `\displaystyle` dans le mode simple, on peut forcer \LaTeX à afficher les mathématiques comme si elles étaient dans le mode mathématique centré.

Exemple :

prenons en particulier `\displaystyle y=\sum_{n=1}^5 a_n` dans l'expression précédente...

donnera

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

La commande `\textstyle` a l'effet inverse.

Les modes mathématiques (4/4)

Avec `\displaystyle` dans le mode simple, on peut forcer \LaTeX à afficher les mathématiques comme si elles étaient dans le mode mathématique centré.

Exemple :

prenons en particulier `\displaystyle y=\sum_{n=1}^5 a_n` dans l'expression précédente...

donnera

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

La commande `\displaystyle` à l'effet inverse.

Les modes mathématiques (4/4)

Avec `\displaystyle` dans le mode simple, on peut forcer \LaTeX à afficher les mathématiques comme si elles étaient dans le mode mathématique centré.

Exemple :

prenons en particulier `\displaystyle y=\sum_{n=1}^5 a_n` dans l'expression précédente...

donnera

prenons en particulier $y = \sum_{n=1}^5 a_n$ dans l'expression précédente...

La commande `\textstyle` à l'effet inverse.

Les symboles mathématiques (1/2)

\times	<code>\times</code>	\subset	<code>\subset</code>	\sim	<code>\sim</code>
\cap	<code>\cap</code>	\subseteq	<code>\subseteq</code>	\approx	<code>\approx</code>
\cup	<code>\cup</code>	\supset	<code>\supset</code>	\neq	<code>\neq</code>
\vee	<code>\lor</code>	\supseteq	<code>\supseteq</code>	\aleph	<code>\aleph</code>
\wedge	<code>\land</code>	\leftarrow	<code>\leftarrow</code>	\Re	<code>\Re</code>
\leq	<code>\leq</code>	\Leftarrow	<code>\Leftarrow</code>	\Im	<code>\Im</code>
\geq	<code>\geq</code>	\rightarrow	<code>\rightarrow</code>	∂	<code>\partial</code>
\leqslant	<code>\leqslant</code>	\Rightarrow	<code>\Rightarrow</code>	\equiv	<code>\equiv</code>
\geqslant	<code>\geqslant</code>	\leftrightarrow	<code>\leftrightarrow</code>	\forall	<code>\forall</code>
\in	<code>\in</code>	\Leftrightarrow	<code>\Leftrightarrow</code>	\exists	<code>\exists</code>
\ni	<code>\ni</code>	\mapsto	<code>\mapsto</code>	\rightarrow	<code>\rightarrow</code>
$\{$	<code>\{</code>	$\}$	<code>\}</code>	$\ $	<code>\ </code>

Les symboles mathématiques (2/2)

Σ	<code>\sum</code>	\cap	<code>\bigcap</code>	$\frac{a}{b}$	<code>\frac{a}{b}</code>
\prod	<code>\prod</code>	\cup	<code>\bigcup</code>	\sqrt{a}	<code>\sqrt{a}</code>
\int	<code>\int</code>	\vee	<code>\bigvee</code>	$\sqrt[n]{a}$	<code>\sqrt[n]{a}</code>
		\wedge	<code>\bigwedge</code>	$\binom{a}{b}$	<code>\binom{a}{b}</code>

Les exposants et les indices

- ^ permet de mettre du texte en exposant et
- _ permet de mettre du texte en indice

Si la partie en exposant ou en indice contient plus d'un caractère, elle doit être comprise entre des accolades. Ces symboles ne peuvent être utilisés qu'en mode mathématique.

Exemple :

on constate aisément que a_n n'affichera pas la même chose que a^n ou que $a^{(n+m)}$ ou bien que a^{n+m} ...

donnera

on constate aisément que a_n n'affichera pas la même chose que a^n ou que a^{n+m} ou bien que $a^n + m$...

Les exposants et les indices

- ^ permet de mettre du texte en exposant et
- _ permet de mettre du texte en indice

Si la partie en exposant ou en indice contient plus d'un caractère, elle doit être comprise entre des accolades. Ces symboles ne peuvent être utilisés qu'en mode mathématique.

Exemple :

on constate aisément que a_n n'affichera pas la même chose que a^n ou que a^{n+m} ou bien que $a^{n+m}...$

donnera

on constate aisément que a_n n'affichera pas la même chose que a^n ou que a^{n+m} ou bien que $a^{n+m}...$

Les exposants et les indices

- ^ permet de mettre du texte en exposant et
- _ permet de mettre du texte en indice

Si la partie en exposant ou en indice contient plus d'un caractère, elle doit être comprise entre des accolades. Ces symboles ne peuvent être utilisés qu'en mode mathématique.

Exemple :

on constate aisément que a_n n'affichera pas la même chose que a^n ou que a^{n+m} ou bien que $a^{n+m} \dots$

donnera

on constate aisément que a_n n'affichera pas la même chose que a^n ou que a^{n+m} ou bien que $a^n + m \dots$

Les fonctions mathématiques

<code>\arccos</code>	<code>\csc</code>	<code>\ker</code>	<code>\min</code>
<code>\arcsin</code>	<code>\deg</code>	<code>\lg</code>	<code>\sec</code>
<code>\arctan</code>	<code>\det</code>	<code>\lim</code>	<code>\sin</code>
<code>\arg</code>	<code>\dim</code>	<code>\liminf</code>	<code>\sinh</code>
<code>\cos</code>	<code>\exp</code>	<code>\limsup</code>	<code>\sup</code>
<code>\cosh</code>	<code>\gcd</code>	<code>\ln</code>	<code>\tan</code>
<code>\cot</code>	<code>\hom</code>	<code>\log</code>	<code>\tanh</code>
<code>\coth</code>	<code>\inf</code>	<code>\max</code>	

Les limites inférieures ou supérieures peuvent également s'écrire avec une barre en-dessous ou au-dessus grâce aux fonctions :

`\varliminf`

`\varlimsup`

Les opérateurs mathématiques (1/2)

Les fonctions mathématiques traditionnelles ont été prédéfinies. Toutefois, il arrive que de nouveaux opérateurs mathématiques apparaissent. L'extension `amsmath` fournit deux commandes permettant de les créer. Leur utilisation doit se faire dans le préambule du document.

`\DeclareMathOperator{<nom de l'opérateur>}{Déclaration}`
`\DeclareMathOperator*{<nom de l'opérateur>}{Déclaration}`

Exemples :

```
\DeclareMathOperator{\mes}{mes}
\mes_n A = 2\mes_{n-1} A

\DeclareMathOperator*{\limex}{\lim\limits_{x \in \mathbb{R}} ex}
K = \lim_{x \in \mathbb{R}} ex |f(x)|
```

Les opérateurs mathématiques (1/2)

Les fonctions mathématiques traditionnelles ont été prédéfinies. Toutefois, il arrive que de nouveaux opérateurs mathématiques apparaissent. L'extension `amsmath` fournit deux commandes permettant de les créer. Leur utilisation doit se faire dans le préambule du document.

```
\DeclareMathOperator{\Nom de l'opérateur}{Déclaration}  
\DeclareMathOperator*{\Nom de l'opérateur}{Déclaration}
```

Exemples

```
\DeclareMathOperator{\mes}{mes}  

$$\text{mes}_n A = 2 \text{mes}_{n-1} A$$
  
\DeclareMathOperator*{\limex}{\lim\limits_{x \in E} ex}  

$$K = \lim_{x \in E} ex |f(x)|$$

```

Les opérateurs mathématiques (1/2)

Les fonctions mathématiques traditionnelles ont été prédéfinies. Toutefois, il arrive que de nouveaux opérateurs mathématiques apparaissent. L'extension `amsmath` fournit deux commandes permettant de les créer. Leur utilisation doit se faire dans le préambule du document.

```
\DeclareMathOperator{\Nom de l'opérateur}{Déclaration}  
\DeclareMathOperator*{\Nom de l'opérateur}{Déclaration}
```

Exemples :

```
\DeclareMathOperator{\mes}{mes}
```

$$\text{mes}_n A = 2 \text{mes}_{n-1} A$$

```
\DeclareMathOperator*{\limex}{\lim\limits_{x \in E} ex}
```

$$K = \lim_{x \in E} ex | f(x) |$$

Les opérateurs mathématiques (1/2)

Les fonctions mathématiques traditionnelles ont été prédéfinies. Toutefois, il arrive que de nouveaux opérateurs mathématiques apparaissent. L'extension `amsmath` fournit deux commandes permettant de les créer. Leur utilisation doit se faire dans le préambule du document.

```
\DeclareMathOperator{\Nom de l'opérateur}{Déclaration}  
\DeclareMathOperator*{\Nom de l'opérateur}{Déclaration}
```

Exemples :

```
\DeclareMathOperator{\mes}{mes}
```

$$\text{mes}_n A = 2 \text{mes}_{n-1} A$$

```
\DeclareMathOperator*{\limex}{\lim\limits_{x \in \mathbb{R}} ex}
```

$$K = \lim_{x \in \mathbb{R}} ex |f(x)|$$

Les opérateurs mathématiques (2/2)

La commande étoilée diffère uniquement de la commande non étoilée en le placement des indices et des exposants par rapport à l'opérateur. Ce placement dépend également du mode mathématique (simple ou centré) utilisé.

L'intérêt de déclarer les nouveaux opérateurs comme tels est double : leur écriture se fera en caractères romains et des espaces appropriés seront ajoutés automatiquement de chaque côté de l'opérateur si nécessaire.

Les opérateurs mathématiques (2/2)

La commande étoilée diffère uniquement de la commande non étoilée en le placement des indices et des exposants par rapport à l'opérateur. Ce placement dépend également du mode mathématique (simple ou centré) utilisé.

L'intérêt de déclarer les nouveaux opérateurs comme tels est double : leur écriture se fera en caractères romains et des espaces appropriés seront ajoutés automatiquement de chaque côté de l'opérateur si nécessaire.

Les alphabets mathématiques

Pour écrire les ensembles \mathbb{N} , \mathbb{Z} , \mathbb{Q} , \mathbb{R} ou \mathbb{C} ,
on tape `\mathbb{b}{...}`

Pour les lettres calligraphiques (\mathcal{A} , \mathcal{B} , ...),
on emploie `\mathcal{...}`.

Pour les lettres gothiques (\mathfrak{A} , \mathfrak{B} , ...),
on écrit `\mathfrak{...}`.

Les lettres grecques (1/2)

Les lettres grecques sont accessibles uniquement en mode mathématique.

α	<code>\alpha</code>	ι	<code>\iota</code>	σ	<code>\sigma</code>
β	<code>\beta</code>	κ	<code>\kappa</code>	ς	<code>\varsigma</code>
γ	<code>\gamma</code>	λ	<code>\lambda</code>	τ	<code>\tau</code>
δ	<code>\delta</code>	μ	<code>\mu</code>	υ	<code>\upsilon</code>
ϵ	<code>\epsilon</code>	ν	<code>\nu</code>	ϕ	<code>\phi</code>
ε	<code>\varepsilon</code>	ξ	<code>\xi</code>	φ	<code>\varphi</code>
ζ	<code>\zeta</code>	π	<code>\pi</code>	χ	<code>\chi</code>
η	<code>\eta</code>	ϖ	<code>\varpi</code>	ψ	<code>\psi</code>
θ	<code>\theta</code>	ρ	<code>\rho</code>	ω	<code>\omega</code>
ϑ	<code>\vartheta</code>	ϱ	<code>\varrho</code>		

Les lettres grecques (2/2)

Le tableau suivant ne représente que les lettres dont l'écriture n'est pas identique à celle d'une lettre romaine.

Γ	<code>\Gamma</code>	Ξ	<code>\Xi</code>	Φ	<code>\Phi</code>
Δ	<code>\Delta</code>	Π	<code>\Pi</code>	Ψ	<code>\Psi</code>
Θ	<code>\Theta</code>	Σ	<code>\Sigma</code>	Ω	<code>\Omega</code>
Λ	<code>\Lambda</code>	Υ	<code>\Upsilon</code>		

Les lettres accentuées et étrangères (1/4)

Certains caractères accentués et liés ne sont pas accessibles directement sur certains claviers. Les règles suivantes permettent d'utiliser ces caractères quel que soit l'environnement.

Les lettres accentuées et étrangères (2/4)

Caractère	Lettre précédée par	Exemple
accent aigu	\ '	é
accent grave	\ `	è
accent circonflexe	\ ^	ê
tréma	\ "	ë
tilde	\ ~	ẽ
macron	\ =	ē
point suscrit	\ .	è

Les lettres accentuées et étrangères (3/4)

Caractère	Commande (lettre entre parenthèses)	Exemple
cédille	<code>\c{...}</code>	ç
brève	<code>\u{...}</code>	ë
hatchek	<code>\v{...}</code>	ě
point souscrit	<code>\d{...}</code>	ẹ
souligné	<code>\b{...}</code>	ē
umlaut	<code>\" {...}</code>	ä
double accent aigu	<code>\H{...}</code>	ú

Les lettres accentuées et étrangères (4/4)

Caractère	Commande	
a rond en chef	<code>\aa</code>	å
ae liés	<code>\ae</code>	æ
l barré	<code>\l</code>	ł
o barré	<code>\o</code>	ø
oe liés	<code>\oe</code>	œ
eszett	<code>\ss</code>	ß

NB. Si le package `babel` n'est pas utilisé, il est conseillé d'utiliser la commande `\i` à la place de la lettre `i` quand celle-ci est accentuée, car cette commande enlèvera le point initial.

Encore plus de symboles...

Scott Pakin : The Comprehensive \LaTeX Symbol list

<http://www.tex.ac.uk/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>

une liste gigantesque de symboles

Sur la page web Detexify

<http://detexify.kirelabs.org/classify.html>

vous pouvez tracer le symbole souhaité, et on vous suggère une commande de \LaTeX

Le retour à la ligne et le changement de page

`\newline` et `\\` permettent le passage à la ligne.

Pour passer à la page suivante, on utilise `\newpage`.

Ces commandes sont à proscrire parce qu'elles imposent une mise en page à L^AT_EX, ce qui n'est pas le but poursuivi.

Le retour à la ligne et le changement de page

`\newline` et `\\` permettent le passage à la ligne.

Pour passer à la page suivante, on utilise `\newpage`.

Attention !

Ces commandes sont à proscrire parce qu'elles imposent une mise en page à \LaTeX , ce qui n'est pas le but poursuivi.

Le commentaire

Dans le code, il suffit d'insérer la commande `%` pour que tout ce qui est écrit sur la ligne après celle-ci ne soit pas pris en considération. Cette partie ne sera visible que dans le code et non dans le document final.

Les polices

<code>\textrm</code>	police roman
<code>\textsf</code>	police sans serif
<code>\texttt</code>	police typewriter
<code>\textbf</code>	série gras
<code>\textit</code>	<i>forme italique</i>
<code>\textsl</code>	<i>forme inclinée</i>
<code>\textsc</code>	FORME PETITES MAJUSCULES
<code>\emph</code>	<i>texte mis en emphase</i>
<code>\underline</code>	<u>texte souligné</u>

exemple :

`\textbf{ce texte est en gras}` produit **ce texte est en gras.**