

Formation L^AT_EX

(d'après des transparents de Denis Bitouzé)

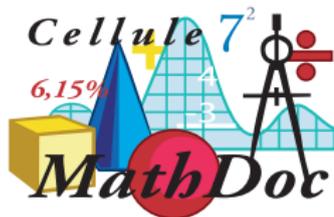
Thierry Bouche

thierry.bouche@ujf-grenoble.fr

Institut Fourier & Cellule MathDoc

<http://www.cedram.org/>

mars 2007





Plan

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

1 Convention

2 Commandes

3 Macros personnelles

4 Modules d'extension

Code source

```
1 \documentclass[12pt]{article}
2 \usepackage[latin1]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{lmodern}
5 \usepackage[a4paper]{geometry}
6 \usepackage[frenchb]{babel}
7 \begin{document}
8 Bonjour monde!
9 \end{document}
```

Convention

Fichier source complet, *préambule* mis en évidence

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Code source

```
1 \documentclass[12pt]{article}
2 \usepackage[latin1]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{lmodern}
5 \usepackage[a4paper]{geometry}
6 \usepackage[frenchb]{babel}
7 \begin{document}
8 Bonjour monde!
9 \end{document}
```

Convention

Fichier source allégé : préambule omis

Code source

```
1  
2  
3  
4  
5  
6  
7 \begin{document}  
8 Bonjour monde!  
9 \end{document}
```

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Convention

Fichier source encore plus allégé : *environnement* **document** omis

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Code source

```
1  
2  
3  
4  
5  
6  
7  
8 Bonjour monde!  
9
```

Code source

```
1 Bonjour monde!
```

signifie

Code source

```
\documentclass[12pt]{article}  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage{lmodern}  
\usepackage[a4paper]{geometry}  
\usepackage[frenchb]{babel}  
\begin{document}  
1 Bonjour monde!  
\end{document}
```

Code source

```
1 Bonjour monde!
```

signifie

Code source

```
\documentclass[12pt]{article}  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage{lmodern}  
\usepackage[a4paper]{geometry}  
\usepackage[frenchb]{babel}  
\begin{document}  
1 Bonjour monde!  
\end{document}
```

Code source

```
1 Bonjour monde!
```

signifie

Code source

```
\documentclass[12pt]{article}  
\usepackage[latin1]{inputenc}  
\usepackage[T1]{fontenc}  
\usepackage{lmodern}  
\usepackage[a4paper]{geometry}  
\usepackage[frenchb]{babel}  
\begin{document}  
1 Bonjour monde!  
\end{document}
```



Plan

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

1 Convention

2 **Commandes**

3 Macros personnelles

4 Modules d'extension



Caractères réservés

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

Certains caractères sont réservés à L^AT_EX et ne peuvent être obtenus directement

Remarque

Par l'artifice des « catcode », tout ceci peut être changé à tout moment !

Remarque

Certains caractères sont réservés à L^AT_EX et ne peuvent être obtenus directement

Remarque

Par l'artifice des « catcode », tout ceci peut être changé à tout moment !

Remarque

Certains caractères sont réservés à L^AT_EX et ne peuvent être obtenus directement

Remarque

Par l'artifice des « catcode », tout ceci peut être changé à tout moment !

Caractères réservés

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Caractère réservé	S'imprime avec
<code>\</code>	<code>\textbackslash</code>
<code>{</code>	<code>\{</code>
<code>}</code>	<code>\}</code>
<code>%</code>	<code>\%</code>
<code>#</code>	<code>\#</code>
<code>\$</code>	<code>\\$</code>
<code>^</code>	<code>\~{ }</code>
<code>~</code>	<code>\~</code>
<code>&</code>	<code>\&</code>
<code>_</code>	<code>_{} </code>

Remarque

- `^` = touche `ALTGr+9` \neq accent circonflexe
- `_` = « underscore » = touche `ALTGr+8`

Caractères réservés

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Caractère réservé	S'imprime avec
<code>\</code>	<code>\textbackslash</code>
<code>{</code>	<code>\{</code>
<code>}</code>	<code>\}</code>
<code>%</code>	<code>\%</code>
<code>#</code>	<code>\#</code>
<code>\$</code>	<code>\\$</code>
<code>^</code>	<code>\~{ }</code>
<code>~</code>	<code>\~</code>
<code>&</code>	<code>\&</code>
<code>_</code>	<code>_ { }</code>

Remarque

- `^` = touche `ALTGr+9` \neq accent circonflexe
- `_` = « underscore » = touche `ALTGr+8`

Propriété

Les commandes (L^A)T_EX sont caractérisées par le fait

- 1 qu'elles commencent par une contre-oblique (\backslash)
- 2 qu'elles sont suivies
 - soit d'un nom constitué exclusivement de lettres
 - soit d'un caractère spécial
- 3 qu'elles se terminent par une espace, un chiffre ou un caractère autre qu'une lettre^a

a. Ce peut être un crochet ouvrant par exemple.

Propriété

Les commandes (L^A)T_EX sont sensibles à la casse : $\backslash AB \neq \backslash Ab$
 $\neq \backslash ab$

Propriété

Les commandes (L^A)T_EX sont caractérisées par le fait

- 1 qu'elles commencent par une contre-oblique (`\`)
- 2 qu'elles sont suivies
 - soit d'un nom constitué exclusivement de lettres
 - soit d'un caractère spécial
- 3 qu'elles se terminent par une espace, un chiffre ou un caractère autre qu'une lettre^a

a. Ce peut être un crochet ouvrant par exemple.

Propriété

Les commandes (L^A)T_EX sont sensibles à la casse : `\AB` \neq `\Ab`
 \neq `\ab`

Propriété

Les commandes (L^A)T_EX sont caractérisées par le fait

- ① qu'elles commencent par une contre-oblique (`\`)
- ② qu'elles sont suivies
 - soit d'un nom constitué exclusivement de lettres
 - soit d'un caractère spécial
- ③ qu'elles se terminent par une espace, un chiffre ou un caractère autre qu'une lettre^a

^a Ce peut être un crochet ouvrant par exemple.

Propriété

Les commandes (L^A)T_EX sont sensibles à la casse : `\AB` \neq `\Ab`
 \neq `\ab`

Propriété

Les commandes (L^A)T_EX sont caractérisées par le fait

- 1 qu'elles commencent par une contre-oblique (\backslash)
- 2 qu'elles sont suivies
 - soit d'un nom constitué exclusivement de lettres
 - soit d'un caractère spécial
- 3 qu'elles se terminent par une espace, un chiffre ou un caractère autre qu'une lettre^a

^a Ce peut être un crochet ouvrant par exemple.

Propriété

Les commandes (L^A)T_EX sont sensibles à la casse : $\backslash AB \neq \backslash Ab$
 $\neq \backslash ab$

Propriété

Les commandes (L^A)T_EX sont caractérisées par le fait

- ① qu'elles commencent par une contre-oblique (\backslash)
- ② qu'elles sont suivies
 - soit d'un nom constitué exclusivement de lettres
 - soit d'un caractère spécial
- ③ qu'elles se terminent par une espace, un chiffre ou un caractère autre qu'une lettre^a

a. Ce peut être un crochet ouvrant par exemple.

Propriété

Les commandes (L^A)T_EX sont sensibles à la casse : $\backslash AB \neq \backslash Ab$
 $\neq \backslash ab$

Propriété

Les commandes (L)A_TE_X sont caractérisées par le fait

- ① qu'elles commencent par une contre-oblique (\backslash)
- ② qu'elles sont suivies
 - soit d'un nom constitué exclusivement de lettres
 - soit d'un caractère spécial
- ③ qu'elles se terminent par une espace, un chiffre ou un caractère autre qu'une lettre^a

a. Ce peut être un crochet ouvrant par exemple.

Propriété

Les commandes (L)A_TE_X sont sensibles à la casse : $\backslash AB \neq \backslash Ab$
 $\neq \backslash ab$

Propriété

Les commandes (L^A)T_EX sont caractérisées par le fait

- ① qu'elles commencent par une contre-oblique (\backslash)
- ② qu'elles sont suivies
 - soit d'un nom constitué exclusivement de lettres
 - soit d'un caractère spécial
- ③ qu'elles se terminent par une espace, un chiffre ou un caractère autre qu'une lettre^a

a. Ce peut être un crochet ouvrant par exemple.

Propriété

Les commandes (L^A)T_EX sont sensibles à la casse : $\backslash AB \neq \backslash Ab$
 $\neq \backslash ab$

Type	Exemple
Commande de caractère	<code>\\$</code>
Commande sans argument	<code>\hfill</code>
Commande avec argument	<code>\textbf{...}</code>
Commande avec arguments ³⁶	<code>\textcolor{green}{...}</code>
Environnement	<code>\begin{center}</code> <code>...</code> <code>\end{center}</code>

Type	Exemple
Commande de caractère	<code>\\$</code>
Commande sans argument	<code>\hfill</code>
Commande avec argument	<code>\textbf{...}</code>
Commande avec arguments ³⁶	<code>\textcolor{green}{...}</code>
Environnement	<code>\begin{center}</code> <code>...</code> <code>\end{center}</code>

Propriété

Une commande est composée de :

- une contre-oblique (`\`)
 - ① au plus un caractère réservé (`\$, \\ _ \. ,` etc.) OU
 - ② une succession quelconque de caractères de type « lettre » (`\alpha, \infty, \documentclass, etc.`)
- Dans le premier cas, la commande est nécessairement interrompue : tout ce qui suit compte (y compris espaces)
- Dans le second cas
 - la commande est interrompue par tout caractère réservé (espace, mais aussi bien parenthèse, chiffre, etc.)
 - L^AT_EX ignore l'espace qui suit la commande

Exemple

Code source

```

1 \begin{itemize}
2 \item \TeX et \LaTeX ,
3 \item \LaTeX2 et \LaTeXe t      ,
4 \item \TeX{} et \LaTeX{} et
5 \item \TeX, \LaTeX{}.
6 \end{itemize}

```

Résultat

- \TeX et \LaTeX ,
- $\LaTeX2$ et $\LaTeX 2_{\epsilon}$,
- \TeX et \LaTeX et
- \TeX, \LaTeX .

Exemple

Code source

```

1 \begin{itemize}
2 \item \TeX et \LaTeX ,
3 \item \LaTeX2 et \LaTeXe t      ,
4 \item \TeX{} et \LaTeX{} et
5 \item \TeX, \LaTeX{}.
6 \end{itemize}

```

Résultat

- T_EX et L^AT_EX,
- L^AT_EX₂ et L^AT_EX_{2 ϵ} t ,
- T_EX et L^AT_EX et
- T_EX, L^AT_EX.

Arguments des commandes (L^A)T_EX

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

De nombreuses commandes (L^A)T_EX prennent des *arguments* qui peuvent être

obligatoires : entre paire(s) d'accolades :

Code source

```
1 \textcolor{blue}{Coucou}
```

optionnels : entre une *unique* paire de crochets :

Code source

```
1 \usepackage[a4paper,landscape]{geometry}
```



Arguments des commandes (L^A)T_EX

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

De nombreuses commandes (L^A)T_EX prennent des *arguments* qui peuvent être

obligatoires : entre paire(s) d'accolades :

Code source

```
1 \textcolor{blue}{Coucou}
```

optionnels : entre une *unique* paire de crochets :

Code source

```
1 \usepackage[a4paper,landscape]{geometry}
```



Arguments des commandes (L^A)T_EX

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

De nombreuses commandes (L^A)T_EX prennent des *arguments* qui peuvent être

obligatoires : entre paire(s) d'accolades :

Code source

```
1 \textcolor{blue}{Coucou}
```

optionnels : entre une *unique* paire de crochets :

Code source

```
1 \usepackage[a4paper,landscape]{geometry}
```

Arguments des commandes (L^A)T_EX

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

De nombreuses commandes (L^A)T_EX prennent des *arguments* qui peuvent être

obligatoires : entre paire(s) d'accolades :

Code source

```
1 \textcolor{blue}{Coucou}
```

optionnels : entre une *unique* paire de crochets :

Code source

```
1 \usepackage[a4paper,landscape]{geometry}
```



Plan

Formation
L^AT_EX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

1 Convention

2 Commandes

3 **Macros personnelles**

4 Modules d'extension

Macros personnelles (sans argument)

ou comment étendre les capacités de \LaTeX

Formation
 \LaTeX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

On peut créer ses propres commandes \LaTeX

Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}{\langle texte \rangle}
```

où :

- $\langle \textit{nom de commande} \rangle$ est un nom de commande au choix :
 - respectant les standards de noms de commandes \LaTeX
 - n'existant ni dans $(\text{\La})\text{\TeX}$ ni dans aucun package chargé
- on n'oubliera pas la contre-oblique devant $\langle \textit{nom de commande} \rangle$
- $\langle \textit{texte} \rangle$ est du texte et/ou code \LaTeX valide

Macros personnelles (sans argument)

ou comment étendre les capacités de \LaTeX

Formation
 \LaTeX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

On peut créer ses propres commandes \LaTeX

Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}{\langle texte \rangle}
```

où :

- $\langle \textit{nom de commande} \rangle$ est un nom de commande au choix :
 - respectant les standards de noms de commandes \LaTeX
 - n'existant ni dans $(\text{\La})\text{\TeX}$ ni dans aucun package chargé
- on n'oubliera pas la contre-oblique devant $\langle \textit{nom de commande} \rangle$
- $\langle \textit{texte} \rangle$ est du texte et/ou code \LaTeX valide

Macros personnelles (sans argument)

ou comment étendre les capacités de \LaTeX

Formation
 \LaTeX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

On peut créer ses propres commandes \LaTeX

Syntaxe

1 `\newcommand{\langle nom de commande \rangle}{\langle texte \rangle}`

où :

- $\langle \textit{nom de commande} \rangle$ est un nom de commande au choix :
 - respectant les standards de noms de commandes \LaTeX
 - n'existant ni dans $(\text{\La})\text{\TeX}$ ni dans aucun package chargé
- on n'oubliera pas la contre-oblique devant $\langle \textit{nom de commande} \rangle$
- $\langle \textit{texte} \rangle$ est du texte et/ou code \LaTeX valide

Macros personnelles (sans argument)

ou comment étendre les capacités de \LaTeX

Formation
 \LaTeX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

On peut créer ses propres commandes \LaTeX

Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}{\langle texte \rangle}
```

où :

- $\langle \textit{nom de commande} \rangle$ est un nom de commande au choix :
 - respectant les standards de noms de commandes \LaTeX
 - n'existant ni dans $(\text{\La})\text{\TeX}$ ni dans aucun package chargé
- on n'oubliera pas la contre-oblique devant $\langle \textit{nom de commande} \rangle$
- $\langle \textit{texte} \rangle$ est du texte et/ou code \LaTeX valide

Macros personnelles (sans argument)

ou comment étendre les capacités de \LaTeX

Formation
 \LaTeX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

On peut créer ses propres commandes \LaTeX

Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}{\langle texte \rangle}
```

où :

- $\langle \textit{nom de commande} \rangle$ est un nom de commande au choix :
 - respectant les standards de noms de commandes \LaTeX
 - n'existant ni dans $(\text{\La})\text{\TeX}$ ni dans aucun package chargé
- on n'oubliera pas la contre-oblique devant $\langle \textit{nom de commande} \rangle$
- $\langle \textit{texte} \rangle$ est du texte et/ou code \LaTeX valide

Macros personnelles (sans argument)

ou comment étendre les capacités de \LaTeX

Formation
 \LaTeX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

On peut créer ses propres commandes \LaTeX

Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}{\langle texte \rangle}
```

où :

- $\langle \textit{nom de commande} \rangle$ est un nom de commande au choix :
 - respectant les standards de noms de commandes \LaTeX
 - n'existant ni dans $(\text{\La})\text{\TeX}$ ni dans aucun package chargé
- on n'oubliera pas la contre-oblique devant $\langle \textit{nom de commande} \rangle$
- $\langle \textit{texte} \rangle$ est du texte et/ou code \LaTeX valide

Macros personnelles (sans argument)

ou comment étendre les capacités de \LaTeX

Formation
 \LaTeX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

On peut créer ses propres commandes \LaTeX

Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}{\langle texte \rangle}
```

où :

- $\langle \textit{nom de commande} \rangle$ est un nom de commande au choix :
 - respectant les standards de noms de commandes \LaTeX
 - n'existant ni dans $(\text{\La})\text{\TeX}$ ni dans aucun package chargé
- on n'oubliera pas la contre-oblique devant $\langle \textit{nom de commande} \rangle$
- $\langle \textit{texte} \rangle$ est du texte et/ou code \LaTeX valide

Macros personnelles (sans argument)

ou comment étendre les capacités de \LaTeX

Formation
 \LaTeX

Thierry
Bouche

Convention

Commandes

Macros
personnelles

Modules
d'extension

Remarque

On peut créer ses propres commandes \LaTeX

Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}{\langle texte \rangle}
```

où :

- $\langle \textit{nom de commande} \rangle$ est un nom de commande au choix :
 - respectant les standards de noms de commandes \LaTeX
 - n'existant ni dans $(\text{\La})\text{\TeX}$ ni dans aucun package chargé
- on n'oubliera pas la contre-oblique devant $\langle \textit{nom de commande} \rangle$
- $\langle \textit{texte} \rangle$ est du texte et/ou code \LaTeX valide

Macros personnelles (sans argument)

Exemples

Exemple

Code source

```
1 Le document que vous allez lire traite de Boris Vian.  
2 Commençons tout de suite par décrire la vie de Boris Vian.  
3 Boris Vian est né le...
```

Résultat

Le document que vous allez lire traite de Boris Vian.
Commençons tout de suite par décrire la vie de Boris Vian.
Boris Vian est né le...

Macros personnelles (sans argument)

Exemples

Exemple

Code source

- ```
1 Le document que vous allez lire traite de Boris Vian.
2 Commençons tout de suite par décrire la vie de Boris Vian.
3 Boris Vian est né le...
```

#### Résultat

Le document que vous allez lire traite de Boris Vian.  
Commençons tout de suite par décrire la vie de Boris Vian.  
Boris Vian est né le...

### Exemple

#### Code source

```
1 \newcommand{\BV}{Boris Vian}
2 Le document que vous allez lire traite de \BV{}.
3 Commençons tout de suite par décrire la vie de \BV{}.
4 \BV{} est né le...
```

#### Résultat

Le document que vous allez lire traite de Boris Vian.  
Commençons tout de suite par décrire la vie de Boris Vian.  
Boris Vian est né le...

# Macros personnelles (sans argument)

## Exemples

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

### Exemple

#### Code source

```
1 \newcommand{\BV}{Boris Vian}
2 Le document que vous allez lire traite de \BV{}.
3 Commençons tout de suite par décrire la vie de \BV{}.
4 \BV{} est né le...
```

#### Résultat

Le document que vous allez lire traite de Boris Vian.  
Commençons tout de suite par décrire la vie de Boris Vian.  
Boris Vian est né le...

# Macros personnelles (sans argument)

## Exemples

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

### Exemple

#### Code source

```
1 \newcommand{\BV}{\textcolor{magenta}{Boris Vian}}
2 Le document que vous allez lire traite de \BV{ }.
3 Commençons tout de suite par décrire la vie de \BV{ }.
4 \BV{ } est né le...
```

#### Résultat

Le document que vous allez lire traite de **Boris Vian**.  
Commençons tout de suite par décrire la vie de **Boris Vian**.  
**Boris Vian** est né le...

# Macros personnelles (sans argument)

## Exemples

### Exemple

#### Code source

```
1 \newcommand{\BV}{\textcolor{magenta}{Boris Vian}}
2 Le document que vous allez lire traite de \BV{ }.
3 Commençons tout de suite par décrire la vie de \BV{ }.
4 \BV{ } est né le...
```

#### Résultat

Le document que vous allez lire traite de **Boris Vian**.  
Commençons tout de suite par décrire la vie de **Boris Vian**.  
**Boris Vian** est né le...

# Macros personnelles (sans argument)

## Exemples

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

### Exemple

#### Code source

```
1 \newcommand{\BV}{\textcolor{magenta}{Jean-Paul Sartre}}
2 Le document que vous allez lire traite de \BV{}.
3 Commençons tout de suite par décrire la vie de \BV{}.
4 \BV{} est né le...
```

#### Résultat

Le document que vous allez lire traite de Jean-Paul Sartre.  
Commençons tout de suite par décrire la vie de Jean-Paul  
Sartre. Jean-Paul Sartre est né le...

# Macros personnelles (sans argument)

## Exemples

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

### Exemple

#### Code source

```
1 \newcommand{\BV}{\textcolor{magenta}{Jean-Paul Sartre}}
2 Le document que vous allez lire traite de \BV{}.
3 Commençons tout de suite par décrire la vie de \BV{}.
4 \BV{} est né le...
```

#### Résultat

Le document que vous allez lire traite de **Jean-Paul Sartre**.  
Commençons tout de suite par décrire la vie de **Jean-Paul Sartre**. **Jean-Paul Sartre** est né le...

L<sup>A</sup>T<sub>E</sub>X connaît trois façons de définir une nouvelle commande :

- `\newcommand` pour les nouvelles macros
- `\renewcommand` pour modifier la définition d'une macro existante

Ne pas modifier le sens d'une commande existante sans savoir ce que l'on fait ! `\H`, `\P`

- `\providecommand` pour fournir une définition par défaut (concerne les programmeurs)

Utilisateurs L<sup>A</sup>T<sub>E</sub>X

Oublier `\def` !

L<sup>A</sup>T<sub>E</sub>X connaît trois façons de définir une nouvelle commande :

- `\newcommand` pour les nouvelles macros
- `\renewcommand` pour modifier la définition d'une macro existante  
 Ne pas modifier le sens d'une commande existante sans savoir ce que l'on fait ! `\H`, `\P`
- `\providecommand` pour fournir une définition par défaut (concerne les programmeurs)

Utilisateurs L<sup>A</sup>T<sub>E</sub>X

Oublier `\def` !

L<sup>A</sup>T<sub>E</sub>X connaît trois façons de définir une nouvelle commande :

- `\newcommand` pour les nouvelles macros
- `\renewcommand` pour modifier la définition d'une macro existante

Ne pas modifier le sens d'une commande existante sans savoir ce que l'on fait ! `\H`, `\P`

- `\providecommand` pour fournir une définition par défaut (concerne les programmeurs)

Utilisateurs L<sup>A</sup>T<sub>E</sub>X

Oublier `\def` !

L<sup>A</sup>T<sub>E</sub>X connaît trois façons de définir une nouvelle commande :

- `\newcommand` pour les nouvelles macros
- `\renewcommand` pour modifier la définition d'une macro existante

**Ne pas modifier le sens d'une commande existante sans savoir ce que l'on fait ! `\H`, `\P`**

- `\providecommand` pour fournir une définition par défaut (concerne les programmeurs)

Utilisateurs L<sup>A</sup>T<sub>E</sub>X

Oublier `\def` !

L<sup>A</sup>T<sub>E</sub>X connaît trois façons de définir une nouvelle commande :

- `\newcommand` pour les nouvelles macros
- `\renewcommand` pour modifier la définition d'une macro existante

**Ne pas modifier le sens d'une commande existante sans savoir ce que l'on fait ! `\H`, `\P`**

- `\providecommand` pour fournir une définition par défaut (concerne les programmeurs)

Utilisateurs L<sup>A</sup>T<sub>E</sub>X

Oublier `\def` !

L<sup>A</sup>T<sub>E</sub>X connaît trois façons de définir une nouvelle commande :

- `\newcommand` pour les nouvelles macros
- `\renewcommand` pour modifier la définition d'une macro existante

Ne pas modifier le sens d'une commande existante sans savoir ce que l'on fait ! `\H`, `\P`

- `\providecommand` pour fournir une définition par défaut (concerne les programmeurs)

Utilisateurs T<sub>E</sub>X

Oublier `\def` !

# Macros personnelles (avec argument(s))

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Propriété

Les nouvelles commandes L<sup>A</sup>T<sub>E</sub>X créées peuvent comporter un ou plusieurs arguments.

Le premier argument peut être optionnel.

# Macros personnelles (avec argument(s))

Exemple : *un* argument

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Exemple

### Code source

```
1 \newcommand{\bli}[1]{\textbf{\textit{#1}}}
2
3 \bli{Roméo} et \bli{Juliette}, couple intemporel...
```

### Résultat

***Roméo*** et ***Juliette***, couple intemporel...

# Macros personnelles (avec argument(s))

Exemple : *un* argument

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Exemple

### Code source

```
1 \newcommand{\bli}[1]{\textbf{\textit{#1}}}
2
3 \bli{Roméo} et \bli{Juliette}, couple intemporel...
```

### Résultat

***Roméo*** et ***Juliette***, couple intemporel...

# Macros personnelles (avec argument(s))

Exemple : *deux* arguments

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Exemple

### Code source

```

1 \newcommand{\bla}[2]{#1 et #2,
2 couple intemporel qui a défié, défie et défiera le temps%
3 }
4 D'abord apparut \bla{Ulysse}{Pénélope}.
5 Ensuite vint \bla{Roméo}{Juliette}.
6 Aujourd'hui, nous confinons à la perfection avec
7 \bla{Johnny}{L\ae{}titia}.

```

### Résultat

D'abord apparut Ulysse et Pénélope, couple intemporel qui a défié, défie et défiera le temps. Ensuite vint Roméo et Juliette, couple intemporel qui a défié, défie et défiera le temps. Aujourd'hui, nous confinons à la perfection avec Johnny et Lætitia, couple intemporel qui a défié, défie et défiera le temps.

# Macros personnelles (avec argument(s))

Exemple : *deux* arguments

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Exemple

### Code source

```

1 \newcommand{\bla}[2]{#1 et #2,
2 couple intemporel qui a défié, défie et défiera le temps%
3 }
4 D'abord apparut \bla{Ulysse}{Pénélope}.
5 Ensuite vint \bla{Roméo}{Juliette}.
6 Aujourd'hui, nous confinons à la perfection avec
7 \bla{Johnny}{L\ae{}titia}.

```

### Résultat

D'abord apparut Ulysse et Pénélope, couple intemporel qui a défié, défie et défiera le temps. Ensuite vint Roméo et Juliette, couple intemporel qui a défié, défie et défiera le temps. Aujourd'hui, nous confinons à la perfection avec Johnny et Lætitia, couple intemporel qui a défié, défie et défiera le temps.

# Macros personnelles (avec argument(s))

Exemple : imbrications de macros

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Exemple

### Code source

```

1 \newcommand{\bli}[1]{\textbf{\textit{#1}}}
2 \newcommand{\bla}[2]{\bli{#1} et \bli{#2},
3 couple intemporel qui a défié, défie et défiera le temps%
4 }
5 D'abord apparut \bla{Ulysse}{Pénélope}.
6 Ensuite vint \bla{Roméo}{Juliette}.
7 Aujourd'hui, nous confinons à la perfection avec
8 \bla{Johnny}{L\ae{}titia}.

```

### Résultat

D'abord apparut *Ulysse* et *Pénélope*, couple intemporel qui a défié, défie et défiera le temps. Ensuite vint *Roméo* et *Juliette*, couple intemporel qui a défié, défie et défiera le temps. Aujourd'hui, nous confinons à la perfection avec *Johnny* et *Lætitia*, couple intemporel qui a défié, défie et défiera le temps.

# Macros personnelles (avec argument(s))

Exemple : imbrications de macros

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Exemple

### Code source

```

1 \newcommand{\bli}[1]{\textbf{\textit{#1}}}
2 \newcommand{\bla}[2]{\bli{#1} et \bli{#2},
3 couple intemporel qui a défié, défie et défiera le temps%
4 }
5 D'abord apparut \bla{Ulysse}{Pénélope}.
6 Ensuite vint \bla{Roméo}{Juliette}.
7 Aujourd'hui, nous confinons à la perfection avec
8 \bla{Johnny}{L\ae{}titia}.

```

### Résultat

D'abord apparut ***Ulysse*** et ***Pénélope***, couple intemporel qui a défié, défie et défiera le temps. Ensuite vint ***Roméo*** et ***Juliette***, couple intemporel qui a défié, défie et défiera le temps. Aujourd'hui, nous confinons à la perfection avec ***Johnny*** et ***Lætitia***, couple intemporel qui a défié, défie et défiera le temps.

# Macros personnelles (avec argument(s))

macro avec argument(s) : syntaxe

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}[\langle nb argument(s) \rangle]{\langle texte \rangle}
```

où

- $1 \leq \langle nb\ argument(s) \rangle \leq 9$
- dans  $\langle texte \rangle$ , l'argument n<sup>o</sup>  $i$  est désigné par  $\#i$

La commande est alors appelée par :

## Syntaxe

```
1 \langle nom de commande \rangle{\langle 1er arg \rangle}{\langle 2e arg \rangle} \cdots {\langle dernier arg \rangle}
```

# Macros personnelles (avec argument(s))

macro avec argument(s) : syntaxe

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}[\langle nb argument(s) \rangle]{\langle texte \rangle}
```

où

- $1 \leq \langle nb\ argument(s) \rangle \leq 9$
- dans  $\langle texte \rangle$ , l'argument n<sup>o</sup>  $i$  est désigné par  $\#i$

La commande est alors appelée par :

## Syntaxe

```
1 \langle nom de commande \rangle{\langle 1er arg \rangle}{\langle 2e arg \rangle} \cdots {\langle dernier arg \rangle}
```

# Macros personnelles (avec argument(s))

macro avec argument(s) : syntaxe

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}[\langle nb argument(s) \rangle]{\langle texte \rangle}
```

où

- $1 \leq \langle nb \text{ argument(s)} \rangle \leq 9$ 
  - dans  $\langle texte \rangle$ , l'argument n<sup>o</sup>  $i$  est désigné par  $\#i$

La commande est alors appelée par :

## Syntaxe

```
1 \langle nom de commande \rangle{\langle 1er arg \rangle}{\langle 2e arg \rangle} \cdots {\langle dernier arg \rangle}
```

# Macros personnelles (avec argument(s))

macro avec argument(s) : syntaxe

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}[\langle nb argument(s) \rangle]{\langle texte \rangle}
```

où

- $1 \leq \langle nb\ argument(s) \rangle \leq 9$
- dans  $\langle texte \rangle$ , l'argument n<sup>o</sup>  $i$  est désigné par  $\#i$

La commande est alors appelée par :

## Syntaxe

```
1 \langle nom de commande \rangle{\langle 1er arg \rangle}{\langle 2e arg \rangle} \dots {\langle dernier arg \rangle}
```

# Macros personnelles (avec argument(s))

macro avec argument(s) : syntaxe

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newcommand{\langle nom de commande \rangle}[\langle nb argument(s) \rangle]{\langle texte \rangle}
```

où

- $1 \leq \langle nb\ argument(s) \rangle \leq 9$
- dans  $\langle texte \rangle$ , l'argument n<sup>o</sup>  $i$  est désigné par  $\#i$

La commande est alors appelée par :

## Syntaxe

```
1 \langle nom de commande \rangle{\langle 1er arg \rangle}{\langle 2e arg \rangle} \cdots {\langle dernier arg \rangle}
```

# Macros personnelles (avec un argument optionnel)

macro avec argument(s) : syntaxe complète

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newcommand{\langle nom \rangle}[\langle nb arg(s) \rangle][\langle défaut 1er arg \rangle]{\langle texte \rangle}
```

où

- $1 \leq \langle nb arg(s) \rangle \leq 9$
- le premier argument est optionnel, sa valeur par défaut est  $\langle défaut 1^{er} arg \rangle$
- dans  $\langle texte \rangle$ , l'argument n<sup>o</sup>  $i$  est désigné par  $\#i$

La commande est alors appelée par :

## Syntaxe

```
1 \langle nom de commande \rangle[\langle 1er arg \rangle]{\langle 2e arg \rangle} \cdots {\langle dernier arg \rangle}
```

# Macros personnelles (avec un argument optionnel)

macro avec argument(s) : syntaxe complète

## Syntaxe

```
1 \newcommand{\langle nom \rangle}[\langle nb arg(s) \rangle][\langle défaut 1er arg \rangle]{\langle texte \rangle}
```

où

- $1 \leq \langle nb arg(s) \rangle \leq 9$
- le premier argument est optionnel, sa valeur par défaut est  $\langle défaut 1^{er} arg \rangle$
- dans  $\langle texte \rangle$ , l'argument n<sup>o</sup>  $i$  est désigné par  $\#i$

La commande est alors appelée par :

## Syntaxe

```
1 \langle nom de commande \rangle[\langle 1er arg \rangle]{\langle 2e arg \rangle} \cdots {\langle dernier arg \rangle}
```

# Macros personnelles (avec un argument optionnel)

macro avec argument(s) : syntaxe complète

## Syntaxe

```
1 \newcommand{\langle nom \rangle}[\langle nb arg(s) \rangle][\langle défaut 1er arg \rangle]{\langle texte \rangle}
```

où

- $1 \leq \langle nb\ arg(s) \rangle \leq 9$
- le premier argument est optionnel, sa valeur par défaut est  $\langle défaut\ 1^{er}\ arg \rangle$
- dans  $\langle texte \rangle$ , l'argument n<sup>o</sup>  $i$  est désigné par  $\#i$

La commande est alors appelée par :

## Syntaxe

```
1 \langle nom de commande \rangle[\langle 1er arg \rangle]{\langle 2e arg \rangle} \cdots {\langle dernier arg \rangle}
```

# Macros personnelles (avec un argument optionnel)

macro avec argument(s) : syntaxe complète

## Syntaxe

```
1 \newcommand{\<nom>}[<nb arg(s)>][<défaut 1er arg>]{<texte>}
```

où

- $1 \leq \langle nb\ arg(s) \rangle \leq 9$
- le premier argument est optionnel, sa valeur par défaut est *défaut 1<sup>er</sup> arg*
- dans *texte*, l'argument n<sup>o</sup> *i* est désigné par #*i*

La commande est alors appelée par :

## Syntaxe

```
1 \<nom de commande>[<1er arg>]{<2e arg>}...{\<dernier arg>}
```

## Syntaxe

```
1 \newcommand{\<nom>}[<nb arg(s)>][<défaut 1er arg>]{<texte>}
```

où

- $1 \leq \langle nb\ arg(s) \rangle \leq 9$
- le premier argument est optionnel, sa valeur par défaut est *⟨défaut 1<sup>er</sup> arg⟩*
- dans *⟨texte⟩*, l'argument n<sup>o</sup> *i* est désigné par #*i*

La commande est alors appelée par :

## Syntaxe

```
1 \<nom de commande>[<1er arg>]{<2e arg>}...{\<dernier arg>}
```

# Environnements personnels (avec argument(s))

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newenvironment{\langle nom d'env \rangle}[\langle nb argument(s) \rangle]
2 {\langle code exécuté au \begin \rangle}
3 {\langle code exécuté au \end \rangle}
```

sous les mêmes conditions qu'un `\newcommand`

La commande est alors appelée par :

## Syntaxe

```
1 \begin{\langle nom d'env. \rangle}{\langle 1er arg \rangle}{\langle 2e g \rangle}...{\langle dernier arg \rangle}
2 ... texte libre...
3 \end{\langle nom d'env. \rangle}
```

Il peut aussi y avoir un premier argument optionnel

# Environnements personnels (avec argument(s))

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newenvironment{\langle nom d'env \rangle}[\langle nb argument(s) \rangle]
2 {\langle code exécuté au \begin \rangle}
3 {\langle code exécuté au \end \rangle}
```

sous les mêmes conditions qu'un `\newcommand`

La commande est alors appelée par :

## Syntaxe

```
1 \begin{\langle nom d'env. \rangle}{\langle 1er arg \rangle}{\langle 2e g \rangle}...{\langle dernier arg \rangle}
2 ... texte libre...
3 \end{\langle nom d'env. \rangle}
```

Il peut aussi y avoir un premier argument optionnel

# Environnements personnels (avec argument(s))

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newenvironment{\langle nom d'env \rangle}[\langle nb argument(s) \rangle]
2 {\langle code exécuté au \begin \rangle}
3 {\langle code exécuté au \end \rangle}
```

sous les mêmes conditions qu'un `\newcommand`

La commande est alors appelée par :

## Syntaxe

```
1 \begin{\langle nom d'env. \rangle}{\langle 1er arg \rangle}{\langle 2e g \rangle}...{\langle dernier arg \rangle}
2 ... texte libre...
3 \end{\langle nom d'env. \rangle}
```

Il peut aussi y avoir un premier argument optionnel

# Environnements personnels (avec argument(s))

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newenvironment{\langle nom d'env \rangle}[\langle nb argument(s) \rangle]
2 {\langle code exécuté au \begin \rangle}
3 {\langle code exécuté au \end \rangle}
```

sous les mêmes conditions qu'un `\newcommand`

La commande est alors appelée par :

## Syntaxe

```
1 \begin{\langle nom d'env. \rangle}{\langle 1er arg \rangle}{\langle 2e g \rangle}...{\langle dernier arg \rangle}
2 ... texte libre...
3 \end{\langle nom d'env. \rangle}
```

Il peut aussi y avoir un premier argument optionnel

# Environnements personnels (avec argument(s))

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Syntaxe

```
1 \newenvironment{\langle nom d'env \rangle}[\langle nb argument(s) \rangle]
2 {\langle code exécuté au \begin \rangle}
3 {\langle code exécuté au \end \rangle}
```

sous les mêmes conditions qu'un `\newcommand`

La commande est alors appelée par :

## Syntaxe

```
1 \begin{\langle nom d'env. \rangle}{\langle 1er arg \rangle}{\langle 2e g \rangle}...{\langle dernier arg \rangle}
2 ... texte libre...
3 \end{\langle nom d'env. \rangle}
```

Il peut aussi y avoir un premier argument optionnel

# Où ranger ses macros ?

- Si elles définissent une nouvelle structure, destinée à être utilisée plusieurs fois (c'est rare !) : créer une *classe* `.cls` (qui peut charger une des classes standard)
- Si c'est un ensemble de macros pratiques que l'on va utiliser dans tous ses articles : créer un *style* `.sty`
- Si elles ne sont destinées qu'à modifier la présentation ou faciliter la saisie d'un document bien précis, les laisser dans le source de ce document

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

# Où ranger ses macros ?

- Si elles définissent une nouvelle structure, destinée à être utilisée plusieurs fois (c'est rare !) : créer une *classe* `.cls` (qui peut charger une des classes standard)
- Si c'est un ensemble de macros pratiques que l'on va utiliser dans tous ses articles : créer un *style* `.sty`
- Si elles ne sont destinées qu'à modifier la présentation ou faciliter la saisie d'un document bien précis, les laisser dans le source de ce document

# Où ranger ses macros ?

- Si elles définissent une nouvelle structure, destinée à être utilisée plusieurs fois (c'est rare!) : créer une *classe* `.cls` (qui peut charger une des classes standard)
- Si c'est un ensemble de macros pratiques que l'on va utiliser dans tous ses articles : créer un *style* `.sty`
- Si elles ne sont destinées qu'à modifier la présentation ou faciliter la saisie d'un document bien précis, les laisser dans le source de ce document



# Où ranger ses macros ?

- Si elles définissent une nouvelle structure, destinée à être utilisée plusieurs fois (c'est rare !) : créer une *classe* `.cls` (qui peut charger une des classes standard)
- Si c'est un ensemble de macros pratiques que l'on va utiliser dans tous ses articles : créer un *style* `.sty`
- Si elles ne sont destinées qu'à modifier la présentation ou faciliter la saisie d'un document bien précis, les laisser dans le source de ce document

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Attention

- Les commandes internes de L<sup>A</sup>T<sub>E</sub>X sont « protégées » par des noms « impossibles » (une fonte peut s'appeler `\T1/cmr/m/n/10.0`; une variable `\@title`)
- Pour modifier la dernière catégorie (niveau petit à moyen gourou), il suffit de le faire dans un `.cls` ou `.sty`; au sein d'un fichier source, il faut insérer la redéfinition dans une paire `\makeatletter-\makeatother` qui annule temporairement le caractère « réservé » du `@`
- Attendre le niveau moyen gourou supérieur avant de toucher à la première !

## Attention

- Les commandes internes de L<sup>A</sup>T<sub>E</sub>X sont « protégées » par des noms « impossibles » (une fonte peut s'appeler `\T1/cmr/m/n/10.0`; une variable `\@title`)
- Pour modifier la dernière catégorie (niveau petit à moyen gourou), il suffit de le faire dans un `.cls` ou `.sty`; au sein d'un fichier source, il faut insérer la redéfinition dans une paire `\makeatletter-\makeatother` qui annule temporairement le caractère « réservé » du `@`
- Attendre le niveau moyen gourou supérieur avant de toucher à la première !

## Attention

- Les commandes internes de L<sup>A</sup>T<sub>E</sub>X sont « protégées » par des noms « impossibles » (une fonte peut s'appeler `\T1/cmr/m/n/10.0`; une variable `\@title`)
- Pour modifier la dernière catégorie (niveau petit à moyen gourou), il suffit de le faire dans un `.cls` ou `.sty`; au sein d'un fichier source, il faut insérer la redéfinition dans une paire `\makeatletter-\makeatother` qui annule temporairement le caractère « réservé » du `@`
- Attendre le niveau moyen gourou supérieur avant de toucher à la première !

## Attention

- Les commandes internes de L<sup>A</sup>T<sub>E</sub>X sont « protégées » par des noms « impossibles » (une fonte peut s'appeler `\T1/cmr/m/n/10.0`; une variable `\@title`)
- Pour modifier la dernière catégorie (niveau petit à moyen gourou), il suffit de le faire dans un `.cls` ou `.sty`; au sein d'un fichier source, il faut insérer la redéfinition dans une paire `\makeatletter-\makeatother` qui annule temporairement le caractère « réservé » du `@`
- Attendre le niveau moyen gourou supérieur avant de toucher à la première !



# Plan

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

1 Convention

2 Commandes

3 Macros personnelles

4 Modules d'extension

## Définition

module = module d'extension<sup>37</sup> = paquet = *package* :  
ensemble de nouvelles définitions et commandes

## Syntaxe (du chargement d'un package)

- 1 `\usepackage{⟨nom du package⟩}` % sans option
- 2 `\usepackage[⟨option(s)⟩]{⟨nom du package⟩}` % avec option(s)

## Remarque

Chargement : uniquement en préambule

## Définition

module = module d'extension<sup>37</sup> = paquet = *package* :  
ensemble de nouvelles définitions et commandes

## Syntaxe (du chargement d'un package)

- 1 `\usepackage{<nom du package>} % sans option`
- 2 `\usepackage[<option(s)>]{<nom du package>} % avec option(s)`

## Remarque

Chargement : uniquement en préambule

## Définition

module = module d'extension<sup>37</sup> = paquet = *package* :  
ensemble de nouvelles définitions et commandes

## Syntaxe (du chargement d'un package)

- 1 `\usepackage{<nom du package>} % sans option`
- 2 `\usepackage[<option(s)>]{<nom du package>} % avec option(s)`

## Remarque

Chargement : uniquement en préambule

## Définition

module = module d'extension<sup>37</sup> = paquet = *package* :  
ensemble de nouvelles définitions et commandes

## Syntaxe (du chargement d'un package)

- 1 `\usepackage{<nom du package>} % sans option`
- 2 `\usepackage[<option(s)>]{<nom du package>} % avec option(s)`

## Remarque

Chargement : uniquement en préambule

## Définition

module = module d'extension<sup>37</sup> = paquet = *package* :  
ensemble de nouvelles définitions et commandes

## Syntaxe (du chargement d'un package)

- 1 `\usepackage{<nom du package>} % sans option`
- 2 `\usepackage[<option(s)>]{<nom du package>} % avec option(s)`

## Remarque

Chargement : uniquement en préambule

# Exemple : dimensions de la page

Package `geometry`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[a4paper]{geometry}
3 ...
4 \begin{document}
5 ...
```

## Remarque

De nombreuses autres options sont disponibles (marges, orientation, etc.)

# Exemple : dimensions de la page

Package `geometry`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[a4paper]{geometry}
3 ...
4 \begin{document}
5 ...
```

## Remarque

De nombreuses autres options sont disponibles (marges, orientation, etc.)

# Exemple : dimensions de la page

Package `geometry`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[a4paper]{geometry}
3 ...
4 \begin{document}
5 ...
```

## Remarque

De nombreuses autres options sont disponibles (marges, orientation, etc.)

# Exemple : adaptation à une (des) langue(s)

Package **babel**

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[frenchb]{babel}
3 ...
4 \begin{document}
5 ...
```

## Remarque

Agit sur les mots clés et date du jour (traduction), usages typographiques, césures, règles de ponctuation, etc.

## Exemple

### Code source

```
1 Document compilé le \today{}
```

### Résultat

Document compilé le  
12 mars 2007

# Exemple : adaptation à une (des) langue(s)

Package `babel`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[frenchb]{babel}
3 ...
4 \begin{document}
5 ...
```

## Remarque

Agit sur les mots clés et date du jour (traduction), usages typographiques, césures, règles de ponctuation, etc.

## Exemple

### Code source

```
1 Document compilé le \today{}
```

### Résultat

Document compilé le  
12 mars 2007

# Exemple : adaptation à une (des) langue(s)

Package **babel**

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[frenchb]{babel}
3 ...
4 \begin{document}
5 ...
```

## Remarque

Agit sur les mots clés et date du jour (traduction), usages typographiques, césures, règles de ponctuation, etc.

## Exemple

### Code source

```
1 Document compilé le \today{}
```

### Résultat

Document compilé le  
12 mars 2007

# Exemple : adaptation à une (des) langue(s)

Package `babel`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[frenchb]{babel}
3 ...
4 \begin{document}
5 ...
```

## Remarque

Agit sur les mots clés et date du jour (traduction), usages typographiques, césures, règles de ponctuation, etc.

## Exemple

### Code source

```
1 Document compilé le \today{}
```

### Résultat

Document compilé le  
12 mars 2007

# Exemple : adaptation à une (des) langue(s)

Package `babel`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[frenchb]{babel}
3 ...
4 \begin{document}
5 ...
```

## Remarque

Agit sur les mots clés et date du jour (traduction), usages typographiques, césures, règles de ponctuation, etc.

## Exemple

### Code source

```
1 Document compilé le \today{}
```

### Résultat

Document compilé le  
12 mars 2007

# Exemple : symbole euro

Package `eurosym`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[right]{eurosym}
3 ...
4 \begin{document}
5 ...
```

## Exemple

### Code source

```
1 \EUR{5}
```

### Résultat

5 €

# Exemple : symbole euro

Package `eurosym`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage[right]{eurosym}
3 ...
4 \begin{document}
5 ...
```

## Exemple

### Code source

```
1 \EUR{5}
```

### Résultat

5 €

# Exemple : symbole euro

Package `eurosym`

## Code source

```
1 ...
2 \usepackage[right]{eurosym}
3 ...
4 \begin{document}
5 ...
```

## Exemple

### Code source

```
1 \EUR{5}
```

### Résultat

5 €

# Exemple : symbole euro

Package `eurosym`

## Code source

```
1 ...
2 \usepackage[right]{eurosym}
3 ...
4 \begin{document}
5 ...
```

## Exemple

### Code source

```
1 \EUR{5}
```

### Résultat

5€

# Exemple : obtenir de la couleur

Package `xcolor`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage{xcolor}
3 ...
4 \begin{document}
5 ...
```

## Exemple

### Code source

```
1 \textcolor{blue!0!red}{Blabla}
2 \textcolor{blue!25!red}{Blabla}
3 \textcolor{blue!50!red}{Blabla}
4 \textcolor{blue!75!red}{Blabla}
5 \textcolor{blue!100!red}{Blabla}
```

### Résultat

Blabla Blabla Blabla  
Blabla Blabla

# Exemple : obtenir de la couleur

Package `xcolor`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage{xcolor}
3 ...
4 \begin{document}
5 ...
```

## Exemple

### Code source

```
1 \textcolor{blue!0!red}{Blabla}
2 \textcolor{blue!25!red}{Blabla}
3 \textcolor{blue!50!red}{Blabla}
4 \textcolor{blue!75!red}{Blabla}
5 \textcolor{blue!100!red}{Blabla}
```

### Résultat

Blabla Blabla Blabla  
Blabla Blabla

## Code source

```
1 ...
2 \usepackage{xcolor}
3 ...
4 \begin{document}
5 ...
```

## Exemple

### Code source

```
1 \textcolor{blue!0!red}{Blabla}
2 \textcolor{blue!25!red}{Blabla}
3 \textcolor{blue!50!red}{Blabla}
4 \textcolor{blue!75!red}{Blabla}
5 \textcolor{blue!100!red}{Blabla}
```

### Résultat

Blabla Blabla Blabla  
Blabla Blabla

# Exemple : obtenir de la couleur

Package `xcolor`

Formation  
L<sup>A</sup>T<sub>E</sub>X

Thierry  
Bouche

Convention

Commandes

Macros  
personnelles

Modules  
d'extension

## Code source

```
1 ...
2 \usepackage{xcolor}
3 ...
4 \begin{document}
5 ...
```

## Exemple

### Code source

```
1 \textcolor{blue!0!red}{Blabla}
2 \textcolor{blue!25!red}{Blabla}
3 \textcolor{blue!50!red}{Blabla}
4 \textcolor{blue!75!red}{Blabla}
5 \textcolor{blue!100!red}{Blabla}
```

### Résultat

Blabla Blabla Blabla  
Blabla Blabla