

Quelques commandes L^AT_EX

Antoine JAMIN

15 décembre 2021

1 Généralités

La déclaration, avant le `\begin{document}`, d'une commande peut vous permettre de déclarer vos propres fonctions pour gérer par exemple d'éventuelle mise en forme automatique.

Une commande se définit de la sorte :

```
\newcommand{nom de la commande}[nombre de paramètres]{code de la commande}
```

Voici par la suite quelques exemples qui peuvent vous aider à créer vos propres commandes.

Quelques liens pour aller plus loin :

- Faire des macros élémentaires (Les tuteurs ENS)
- L^AT_EX créer ses commandes (cours de Bertrand Masson) [PDF]
- L^AT_EX/Macros (wikibooks)

2 Quelques exemples simples que j'utilise régulièrement

Author *et al.*

Commande

```
\newcommand{\etal}[1]{#1 \textit{et al.}}
```

Utilisation et résultat

```
\etal{Shannon} → Shannon et al.
```

...×10^{...}

Commandes

```
\newcommand{\tenpow}[2]{\ensuremath{\{#1\}\times 10^{\{#2\}}}  
\newcommand{\tenpowm}[2]{\ensuremath{\{#1\}\times 10^{-\{#2\}}}  
\newcommand{\tenpowp}[2]{\ensuremath{\{#1\}\times 10^{+\{#2\}}}
```

Utilisations et résultats

```
\tenpow{a}{i} → a×10i  
\tenpowm{a}{i} → a×10-i  
\tenpowp{a}{i} → a×10+i
```

Modification dans le texte

Commandes

```
\usepackage[dvipsnames]{xcolor}  
\xdefinecolor{mgreen}{named}{Green}  
\usepackage{ulem}  
\newcommand{\modifAdd}[1]{\textcolor{mgreen}{#1}}  
\newcommand{\modifRem}[1]{\textcolor{red}{\sout{#1}}}
```

Utilisations et résultats

```
\modifAdd{J'ajoute du texte.} → J'ajoute du texte.  
\modifRem{Je supprime du texte.} → Je supprime du texte.
```

Formatage des textes facile à modifier

Lors de l'écriture d'un rapport ou d'une lettre vous aurez sûrement besoin de gérer la mise en valeur de texte et ou de mots anglais dans un texte en français. Afin de pouvoir gérer le format de cette mise en forme pour tous les textes et pouvoir la modifier sans avoir à changer partout dans votre code (seulement la ligne de déclaration de commande devra être modifiée) voici un exemple de deux commandes : `\mev` permettant de mettre en valeur un mot et `\en` permettant de formater les mots anglais.

Commandes

```
\newcommand{\mev}[1]{\textbf{\huge{#1}}}  
\newcommand{\en}[1]{\emph{#1}}
```

Utilisations et résultats

Je met ce `\mev{mot}` ou `\mev{ces mots}` en valeur. → Je met ce **mot** ou **ces mots** en valeur.

Les méthodes d'entropie croisée (`\en{cross-entropy}`). → Les méthodes d'entropie croisée (*cross-entropy*).

Pour aller plus loin

Si vous souhaitez modifier la mise en forme il vous suffira d'adapter le contenu des accolades après [1] dans la commande. Vous pouvez également créer vos propres fonctions en définissant leur nom dans les premières accolades de la commande et en déclarant la mise en forme dans les dernières accolades de la commandes.

3 Usage avancé : conditions

Il est possible d'utiliser des conditions à l'intérieur de vos commandes dans le but, par exemple, d'adapter une mise en forme en fonction du contenu de celle-ci.

A l'intérieur de la définition de la fonction (après [·]) il est possible d'utiliser les conditions primitives de L^AT_EX à savoir : `\if`, `\ifnum`, `\ifdim`, `\else`, `\fi`, ...

Il est également possible d'utiliser le package `ifthen`.

Illustration par l'exemple : *p*-value

Dans cet exemple nous allons développer une fonction permettant de mettre en valeur les valeurs de *p*-value significatives (inférieures à 5×10^{-2}). Pour ce faire nous allons utiliser les fonctions `\tenpowm` et `\mev` présentées précédemment.

Commande

L'algorithme sera le suivant :

- Si le second argument (puissance de 10) est inférieur à 2 : écriture de la *p*-value sans mise en valeur.
- Sinon :
 - Si le second argument (puissance de 10) est égal à 2 :
 - Si le premier argument (nombre devant la puissance de 10) est inférieur à 5 : écriture de la *p*-value avec mise en valeur.
 - Sinon : écriture de la *p*-value sans mise en valeur.
 - Sinon : écriture de la *p*-value avec mise en valeur.

Avec les conditions primitives de L^AT_EX :

```
\newcommand{\pvalue}[2]{\ifnum#2<2 {\tenpowm{#1}{#2}}\
  else{\ifnum#2=2 {\ifdim#1pt<5pt {\mev{\tenpowm
    {#1}{#2}}}\else{\tenpowm{#1}{#2}}\fi}\else{\mev{\
    tenpowm{#1}{#2}}}\fi}\fi}}
```

Avec le package `ifthen` :

```
\usepackage{ifthen}
\newcommand{\pvalue}[2]{\ifthenelse{#2<2}{\tenpowm
  {#1}{#2}}{\ifthenelse{#2=2}{\ifthenelse{#1<5}{\mev{\
    tenpowm{#1}{#2}}}\tenpowm{#1}{#2}}{\mev{\
    tenpowm{#1}{#2}}}}
```

Utilisation et résultat

`\pvalue{3}{1}` → 3×10^{-1}

`\pvalue{5}{2}` → 5×10^{-2}

`\pvalue{4}{2}` → 4×10^{-2}

`\pvalue{8}{4}` → 6×10^{-4}