

Cahiers **GUT**enberg

☞ PERSONNALISER LES LISTES L^AT_EX
☞ Michel GOOSSENS

Cahiers GUTenberg, n° 17 (1994), p. 32-47.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1994__17_32_0>

© Association GUTenberg, 1994, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

Toute utilisation commerciale ou impression systématique

est constitutive d'une infraction pénale. Toute copie ou impression

de ce fichier doit contenir la présente mention de copyright.

Personnaliser les listes L^AT_EX

Michel GOOSSENS

CERN

Division CN

CH-1211 Genève 23

Suisse

E-mail : michel.goossens@cern.ch

Résumé. Cet article est une adaptation pour les *Cahiers* d'une partie du troisième chapitre du livre « The L^AT_EX Companion »¹. Tout d'abord les différents paramètres et commandes qui contrôlent les listes L^AT_EX usuelles, `enumerate`, `itemize`, et `description` sont passés en revue. Ensuite, l'environnement généralisé `list` est introduit et il est montré comment vous pouvez personnaliser sa présentation en modifiant la valeur des paramètres de style qui contrôlent cet environnement.

Abstract. *In this article², which is a slightly reworked version of part of the third chapter of the book "The L^AT_EX Companion" (Addison-Wesley, January 1994) we take a look at L^AT_EX's list environments. First the various parameters and commands controlling the standard L^AT_EX lists, `enumerate`, `itemize`, and `description` are discussed. Then the general `list` environment is introduced and we tell you how to build custom layouts by varying the values of the parameters controlling this environment.*

1. Modifier les listes L^AT_EX

Les listes sont une construction très générale en L^AT_EX et elles sont utilisées pour composer plusieurs des environnements de mise en évidence de L^AT_EX. Dans cette section les trois listes standard L^AT_EX sont décrites, c'est-à-dire les environnements `enumerate`, `itemize` et `description`. La liste généralisée sera le sujet de la section 2 à la page 40.

Il est relativement aisé de personnaliser les trois listes standard de L^AT_EX, et chacune d'elles sera passée en revue ci-dessous. On peut modifier

1. Ce livre est édité chez Addison-Wesley, et sa sortie a eu lieu en janvier 1994. Les auteurs sont Michel Goossens, Frank Mittelbach et Alexander Samarin. Une description détaillée de son contenu a paru dans *La Lettre GUTenberg* numéro 2, pages 4–6. Notons que, pour aider à financer le projet L^AT_EX3, la moitié des droits d'auteurs du livre sera versée directement à ce projet.

2. This article was published in English in the magazines of the Dutch and Polish T_EX User Groups

globalement la définition par défaut de ces environnements en redéfinissant dans le préambule certains des paramètres qui contrôlent la présentation de la liste en question. Alternativement on peut choisir une modification avec une portée limitée.

1.1. Personnaliser la liste `enumerate`

La liste L^AT_EX qui permet d'énumérer un nombre d'éléments est construite à l'aide de l'environnement `enumerate`. Cet environnement est décrit par les commandes que vous trouverez dans le tableau 1 à la page 38. La première ligne spécifie le nom des compteurs utilisés pour les quatre niveaux de la liste. Les deuxième et troisième lignes sont, respectivement, les commandes et leurs définitions par défaut, définissant la représentation de ces compteurs dans les classes³ standard de L^AT_EX. Les lignes quatre, cinq et six contiennent, respectivement, les commandes, leurs définitions par défaut et des exemples pour le champ numéroté de la liste.

Une référence à un élément d'une liste ordonnée est composée à partir d'une des commandes `\theenumi`, `\theenumii`, etc., préfixée d'une commande `\p@enumi`, `\p@enumii`, etc., respectivement. Les trois dernières lignes du tableau montrent ces commandes, leurs définitions par défaut, et des exemples de la représentation d'une référence dans le texte. Il est important que vous définissiez correctement les deux commandes qui caractérisent une liste ordonnée, c'est-à-dire la présentation typographique des éléments de la liste et celle des références dans le texte à un des éléments.

Avec ces connaissances nouvellement acquises, vous pouvez maintenant construire différents types de listes ordonnées.

Un premier exemple redéfinit la présentation des compteurs du premier et deuxième niveau pour qu'ils utilisent des chiffres romains majuscules et des lettres latines. En plus, le champ numéroté de la liste est composé du compteur en question suivi d'un point. Pour la définition des commandes des préfixes des références, je me suis limité à une redéfinition pour le

3. L^AT_EX2_ε, la version mise-à-jour de L^AT_EX fait la distinction entre *classes* et *paquets*. Les documents dans une même classe sont caractérisés par une structure générique commune. La classe à laquelle un document appartient est spécifiée par la commande `\documentclass`. Des exemples de classes sont `article`, `report` et `book`. Si vous voulez préciser certains aspects globaux du document (typographiques, comme le corps de la police, ou la langue principale) vous pouvez spécifier des « options de classe » comme paramètres optionnels entre crochets, par exemple `\documentclass[11pt,french]{article}`. D'un autre côté, les paquets regroupent la définition de commandes et environnements qui augmentent les fonctionnalités de base de L^AT_EX. Ils sont déclarés par une commande `\usepackage`, comme par exemple `\usepackage{multicol}`.

deuxième niveau. La commande `\p@enumi` utilise la valeur par défaut du tableau 1 à la page 38.

```
\makeatletter
\renewcommand{\theenumi}{\Roman{enumi}}
\renewcommand{\labelenumi}{\theenumi.}
\renewcommand{\theenumii}{\Alph{enumii}}
\renewcommand{\labelenumii}{\theenumii.}
\renewcommand{\p@enumii}{\theenumi.}
\makeatother
\begin{enumerate} \item \textbf{L'Amérique}
  \begin{enumerate}
    \item \textbf{Les États Unis} \label{q1}
      Les cinquante états, leur population, caractéristiques
      géographiques et importance économique. \label{q1}
    \item \textbf{Le Canada} \label{q2}
      Les dix provinces canadiennes, leur géographie, histoire,
      et économie. \label{q2}
  \end{enumerate}
  \item \textbf{L'Asie} \label{q3}
  \item \textbf{L'Europe} \label{q4}
\end{enumerate}
q1=\ref{q1} q2=\ref{q2} q3=\ref{q3} q4=\ref{q4}
```

I. L'Amérique

A. Les États Unis

Les cinquante états, leur population, caractéristiques géographiques et importance économique.

B. Le Canada

Les dix provinces canadiennes, leur géographie, histoire, et économie.

II. L'Asie

III. L'Europe

q1=I.A q2=I.B q3=II q4=III

Vous pouvez également décorer le champ numéroté de la liste en y introduisant un caractère de votre choix. Dans l'exemple ci-dessous le symbole « paragraphe » (`\S`) est utilisé pour préfixer chaque numéro de la liste au premier niveau.

```
\renewcommand{\labelenumi}{\S\theenumi.}
\begin{enumerate}
\item texte de la liste, encore du texte dans la liste ; \label{w1}
\item texte de la liste, encore du texte dans la liste ; \label{w2}
\item texte de la liste, encore du texte dans la liste.
\end{enumerate}
w1=\ref{w1} w2=\ref{w2}
```

- §I. texte de la liste, encore du texte dans la liste ;
- §II. texte de la liste, encore du texte dans la liste ;
- §III. texte de la liste, encore du texte dans la liste.

w1=I w2=II

Peut-être voudrez-vous utiliser des marques différentes pour numéroter chaque élément consécutif de la liste. Dans l'exemple suivant, des caractères de la police PostScript *ZapfDingbats* sont utilisés. Dans ce cas il n'y a pas de moyen simple pour que la commande `\ref` produise une référence correcte. Vous pourrez néanmoins utiliser l'environnement `dingautolist` disponible dans le paquet `pifont`, qui fait partie du système `PSNFSS`. Notez également que l'exemple utilise les extensions du paquet `calc` pour faire l'addition à l'intérieur de la commande `\setcounter`.

```
\newcounter{local}
\renewcommand{\labelenumi}{%
  {\setcounter{local}{171+\value{enumi}}\ding{\value{local}}}
\begin{enumerate}
\item texte de la liste, et encore du texte dans la liste ;
\item texte de la liste, et encore du texte dans la liste ;
\item texte de la liste, et encore du texte dans la liste.
\end{enumerate}
```

- ① texte de la liste, et encore du texte dans la liste ;
- ② texte de la liste, et encore du texte dans la liste ;
- ③ texte de la liste, et encore du texte dans la liste.

Enfin, pour ceux qui n'ont pas envie de bricoler, et qui préfèrent les solutions toutes faites, il existe le paquet `enumerate` (de David Carlisle), qui redéfinit l'environnement `enumerate`, en y ajoutant un argument facultatif, qui vous permet de spécifier le style à utiliser pour représenter le compteur

de la liste. L'argument facultatif peut contenir un des symboles A, a, I, i, ou 1 qui représentent, respectivement, les styles de représentation `\Alph`, `\alph`, `\Roman`, `\roman`, et `\arabic` pour le compteur.

En plus, vous pouvez entourer ces symboles avec une suite de caractères de votre choix, qui peut contenir des expressions `TeX`; cependant les caractères A, a, I, i, et 1 doivent être spécifiés à l'intérieur d'un groupe `{}` si vous ne voulez pas qu'ils soient interprétés comme décrit ci-dessus.

Les commandes pour construire les références `\label` et `\ref` fonctionnent également avec le paquet `enumerate`. Il faut noter, cependant, que seul le style de représentation du compteur, comme défini par la présence d'un des caractères A, a, I, i, ou 1 dans l'argument facultatif de l'environnement, sera utilisé dans la construction de la référence, et non pas l'argument complet.

```
\begin{enumerate}[Ex i.]
\item texte du premier élément au niveau un.
    Encore du texte du premier élément au niveau un.      \label{LA}
\item texte du deuxième élément au niveau un.
    \begin{enumerate}[point a]
    \item texte du premier élément au niveau deux.
        Encore du texte du premier élément au niveau deux.  \label{LB}
    \item texte du deuxième élément au niveau deux.
    \end{enumerate}
\end{enumerate}
\begin{enumerate}[{A}-1]
\item texte du premier élément de la deuxième liste.
    Encore du texte du premier élément de la deuxième liste.
                                                \label{LC}
\item texte du deuxième élément de la deuxième liste.
\end{enumerate}
Et voici comment vous référencez les éléments de ces listes.
\og\ref{LA}\fg, \og\ref{LB}\fg{} et \og\ref{LC}\fg{} ou
plus correctement \og{Ex~}\ref{LA}.\fg{} et \og{A-}\ref{LC}\fg.
```

Cet exemple génère ce qui suit :

Ex i. texte du premier élément au niveau un. Encore du texte du premier élément au niveau un.

Ex ii. texte du deuxième élément au niveau un.

point a) texte du premier élément au niveau deux. Encore du texte du premier élément au niveau deux.

point b) texte du deuxième élément au niveau deux.

A-1 texte du premier élément de la deuxième liste. Encore du texte du premier élément de la deuxième liste.

A-2 texte du deuxième élément de la deuxième liste.

Et voici comment vous référencez les éléments de ces listes. « i », « ii.a » et « 1 » ou plus correctement « Ex i. » et « A-1 ».

1.2. Personnaliser la liste `itemize`

Les commandes contrôlant les listes non-ordonnées définies par l'environnement `itemize` sont présentées dans le tableau 2 à la page 38.

Si vous voulez créer une liste avec d'autres étiquettes, vous pouvez redéfinir la commande qui génère cette dernière. Vous pouvez faire ce changement localement, pour une liste, comme le montre l'exemple ci-dessous, ou vous pouvez décider que toutes vos listes non-ordonnées utiliseront la même représentation en introduisant la redéfinition en question dans le préambule de votre document, c'est-à-dire entre les commandes `\documentclass` et `\begin{document}`. La liste non-ordonnée présentée dans l'exemple suivant utilise un symbole de la police PostScript *ZapfDingbats*⁴ comme étiquette au premier niveau.

```
\newenvironment{MYitemize}%
  {\renewcommand{\labelitemi}{\ding{43}}\begin{itemize}}%
  {\end{itemize}}
\begin{MYitemize}
\item Texte du premier élément de la liste.
\item Le deuxième élément de la liste.
\item Cette phrase est le troisième élément de la liste.
\end{MYitemize}
```

Ce qui génère le résultat suivant :

- ☞ Texte du premier élément de la liste.
- ☞ Le deuxième élément de la liste.
- ☞ Cette phrase est le troisième élément de la liste.

4. La commande `\ding` est définie dans le paquet `pifont` et fait partie du système PSNFSS.

	niveau un	niveau deux	niveau trois	niveau quatre
<i>compteur</i>	<code>enumi</code>	<code>enumii</code>	<code>enumiii</code>	<code>enumiv</code>
<i>représentation</i>	<code>\theenumi</code>	<code>\theenumii</code>	<code>\theenumiii</code>	<code>\theenumiv</code>
<i>définition par défaut</i>	<code>\arabic{enumi}</code>	<code>\alph{enumii}</code>	<code>\roman{enumiii}</code>	<code>\Alph{enumiv}</code>
<i>champ numéroté</i>	<code>\labelenumi</code>	<code>\labelenumii</code>	<code>\labelenumiii</code>	<code>\labelenumiv</code>
<i>définition par défaut</i>	<code>\theenumi.</code>	<code>(\theenumii)</code>	<code>\theenumiii.</code>	<code>\theenumiv.</code>
<i>exemple</i>	1., 2.	(a), (b)	i, ii.	A, B.
<i>préfixe</i>	<code>\p@enumi</code>	<code>\p@enumii</code>	<code>\p@enumiii</code>	<code>\p@enumiv</code>
<i>définition par défaut</i>	<code>{}</code>	<code>\theenumi</code>	<code>\theenumi(\theenumii)</code>	<code>\p@enumiii\theenumiii</code>
<i>exemple d'une référence</i>	1, 2	1a, 2b	1(a)i, 2(b)ii	1(a)iA, 2(b)iiB

TAB.1 Commandes contrôlant la présentation d'une liste ordonnée (*environnement enumerate*)

	niveau un	niveau deux	niveau trois	niveau quatre
<i>commande</i>	<code>\labelitemi</code>	<code>\labelitemii</code>	<code>\labelitemiii</code>	<code>\labelitemiv</code>
<i>définition</i>	<code>\$\$\bullet\$\$</code>	<code>\bf --</code>	<code>\$\$\ast\$\$</code>	<code>\$\$\cdot\$\$</code>
<i>représentation</i>	•	—	*	.

Il est à noter que la typographie française n'utilise pas les étiquettes par défaut, ce sont des tirets courts (–) qui sont utilisés à tous les niveaux.

TAB.2 Commandes contrôlant la présentation d'une liste non-ordonnée (*environnement itemize*)

1.3. Personnaliser la liste `description`

L'environnement `description` vous offre la possibilité de personnaliser la commande `\descriptionlabel`, qui génère le terme de l'élément de la liste de définition. Dans l'exemple ci-dessous la police utilisée pour composer le terme de la liste est changée de « gras » (le défaut) en « petites capitales ».

```
\renewcommand{\descriptionlabel}[1]{\hspace{\labelsep}\textsc{#1}}
\begin{description}
\item[A.] texte dans une liste, encore du texte dans une liste,
           texte dans une liste, encore du texte dans une liste ;
\item[B.] texte dans une liste, encore du texte dans une liste,
           texte dans une liste, encore du texte dans une liste ;
\item[C.] texte dans une liste, encore du texte dans une liste,
           texte dans une liste, encore du texte dans une liste.
\end{description}
```

Ce qui donne :

- A. texte dans une liste, encore du texte dans une liste, texte dans une liste, encore du texte dans une liste ;
- B. texte dans une liste, encore du texte dans une liste, texte dans une liste, encore du texte dans une liste ;
- C. texte dans une liste, encore du texte dans une liste, texte dans une liste, encore du texte dans une liste.

Pour bien comprendre la redéfinition de la commande `\descriptionlabel` ci-dessus, il faut savoir que les classes de document standard *L^AT_EX* positionnent le point de départ de la boîte qui contient le terme de définition dans un environnement `description` à une distance de `-\labelsep` à gauche de la marge de gauche de l'environnement englobant. Ainsi, la commande `\descriptionlabel` doit d'abord ajouter une valeur de `\labelsep` pour aligner le début du terme avec la marge de gauche.

2. Comment construire ses propres listes

Les listes sont construites à partir de l'environnement générique L^AT_EX `list`, comme suit :

```
\begin{list}{étiquette_par_défaut}{déclarations}
  éléments_de_la_liste
\end{list}
```

Le paramètre *étiquette_par_défaut* spécifie le texte à utiliser comme étiquette des éléments de la liste quand la commande `\item` est spécifiée sans argument facultatif. Le paramètre *déclarations* définit les différents paramètres géométriques caractérisant l'environnement `list` (voyez la figure 1 à la page 41). Ces paramètres de styles définis dans cette figure, ainsi que leurs valeurs par défaut sont décrits plus en détail ci-dessous. Ils peuvent tous être redéfinis à l'aide des commandes `\setlength` et `\addtolength`.

Dimensions verticales

Toutes les dimensions verticales sont de type élastique avec une valeur qui dépend du corps de la police et du niveau de la liste.

`\topsep` Espacement entre le premier élément et le paragraphe précédant la liste.

`\partopsep` Espace supplémentaire ajouté à `\topsep` quand l'environnement commence un nouveau paragraphe.

`\itemsep` Espacement entre deux éléments consécutifs.

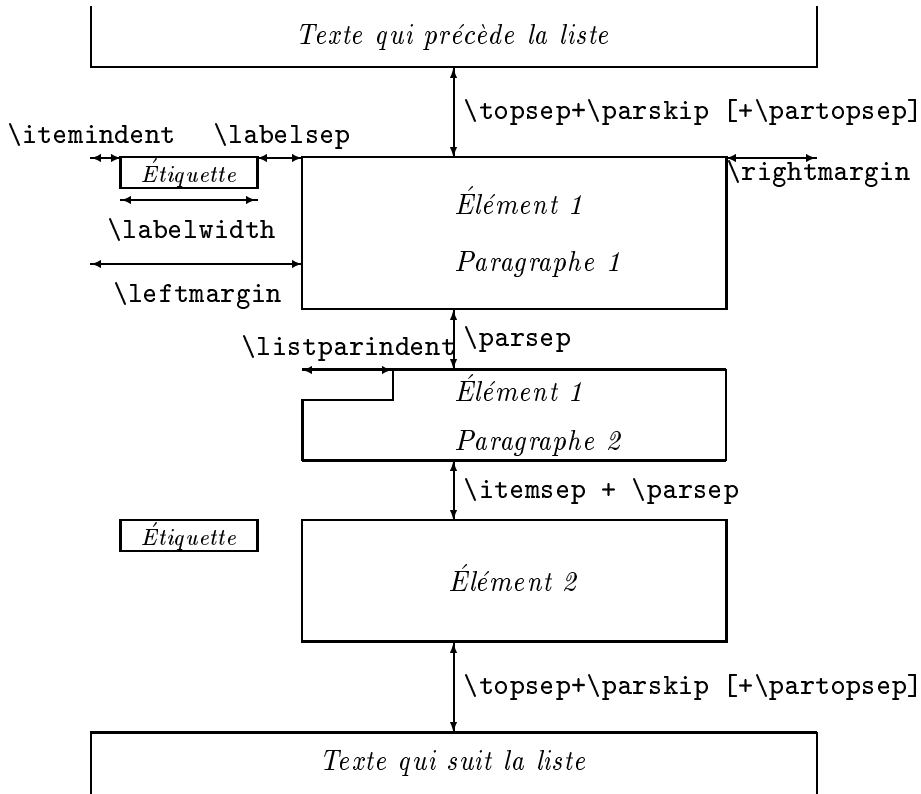
`\parsep` Espacement entre deux paragraphes d'un même élément.

Dimensions horizontales

`\leftmargin` Distance entre la marge de gauche de l'environnement entourant (ou de la page, s'il s'agit d'une liste du premier niveau) et la marge de gauche de la liste. Sa valeur, qui doit être positive, dépend du niveau de la liste.

`\rightmargin` Comme `\leftmargin`, mais pour la marge de droite. Sa valeur par défaut est 0 pt.

`\listparindent` Renforcement supplémentaire au début de chaque paragraphe de la liste, à l'exception du premier, qui suit la commande `\item`. Sa valeur, qui est 0 pt par défaut, peut être négative.


 FIGURE 1 - Les commandes contrôlant la structure d'une liste *L^AT_EX*

`\itemindent` Renforcement supplémentaire devant l'étiquette des éléments d'une liste. Sa valeur par défaut est 0 pt.

`\labelwidth` La largeur nominale de la boîte contenant l'étiquette. Si la largeur naturelle de l'étiquette est inférieure ou égale à `\labelwidth`, alors l'étiquette sera appuyée à droite dans une boîte de largeur `\labelwidth`. Dans le cas contraire, une boîte d'une largeur naturelle est construite, ce qui a comme effet de renfoncer le texte sur la même ligne vers la droite.

`\labelsep` La distance entre la fin de la boîte contenant l'étiquette et le texte de l'élément. Sa valeur, qui est 0,5 em par défaut, peut être négative.

Plusieurs environnements *L^AT_EX* sont définis à l'aide de `list` (par exemple `quote`, `quotation`, `center`, `flushleft`, et `flushright`). Remar-

quez que ces environnements ont seulement un élément et que la commande `\item[]` est incluse dans la définition de l'environnement.

Pour illustrer ceci, considérons l'environnement `quote`, qui est défini avec des marges gauches et droites identiques. Une variante simple, `Quote`, est présentée ci-dessous. Elle est identique à `quote`, mais ajoute des guillemets autour du texte. Il faut aussi prendre quelques précautions pour s'assurer que des blancs parasites ne viennent pas s'insérer avant (`\ignorespaces`) ou après (`\unskip`) le texte.

```
\newenvironment{Quote}% Définition de Quote
  {\begin{list}{}{%
    \setlength{\rightmargin}{\leftmargin}}\item[]\og\ignorespaces}
  {\unskip\fg\end{list}}
\ldots\ texte précédant la liste.
\begin{Quote}
  Le texte de la citation. Encore du texte pour la citation.
  Le texte de la citation. Encore du texte pour la citation.
\end{Quote}
Texte suivant la liste\ldots
```

... texte précédant la liste.

« Le texte de la citation. Encore du texte pour la citation. Le
texte de la citation. Encore du texte pour la citation. »

Texte suivant la liste...

Des listes généralisées peuvent être utiles pour documenter des commandes d'ordinateur ou les fonctions d'un programme. On trouvera une illustration de cette fonctionnalité dans la suite d'exemples ci-dessous, qui utilisent l'environnement `entry` et ses variantes. Dans chaque cas le nom de l'élément décrit est entré entre crochets comme paramètre facultatif de la commande `\item`.

La liste suivante contient une redéfinition de la commande `\makelabel` et des dimensions horizontales `\labelwidth` et `\leftmargin`⁵.

```
\newcommand{\entrylabel}[1]{\mbox{\textsf{#1 :}}\hfil}
\newenvironment{entry}%
  {\begin{list}{}{\renewcommand{\makelabel}{\entrylabel}%
    \setlength{\labelwidth}{35pt}%
    \setlength{\leftmargin}{\labelwidth+\labelsep}}}%
```

5. Cet exemple et les suivants nécessitent les paquets `calc` et `ifthen`.

```

{\end{list}}

\begin{entry}
\item[Édition]
  Tout travail de composition d'un document commence par l'introduction
  du texte \a saisir dans l'ordinateur \a l'aide d'un éditeur.
\item[Compilation] En utilisant \LaTeX{}.
\item[Correction des erreurs]\mbox{}\\
  Les éventuelles erreurs sont corrigées;
  on recompile ensuite le document.
\end{entry}

```

Édition : Tout travail de composition d'un document commence par l'introduction du texte à saisir dans l'ordinateur à l'aide d'un éditeur.

Compilation : En utilisant L^AT_EX.

Correction des erreurs :

Les éventuelles erreurs sont corrigées ; on recompile ensuite le document.

Cet exemple montre un problème typique que l'on rencontre avec des listes de type description, lorsque le texte du *terme* de la liste est plus large que la largeur de l'étiquette. Par défaut, L^AT_EX continue le texte du terme dans la partie *description* de la liste. Souvent ce comportement n'est pas opportun et pour des raisons d'équilibre visuel il est plus astucieux de commencer la partie descriptive de l'élément de la liste sur la ligne suivante. Ici, une nouvelle ligne était forcée en introduisant une boîte vide suivi de la commande « `\` ».

Dans ce qui suit plusieurs possibilités pour contrôler la largeur et le positionnement mutuel des parties terme et description de la liste seront présentées. Une première méthode consiste à changer la largeur du terme. L'environnement `Ventry` a un argument, qui permet de spécifier la largeur désirée pour le champ du terme (qui correspond habituellement au texte du terme le plus large). Remarquez la redéfinition de la commande `\makeLabel` qui contrôle la présentation typographique du terme. Comme cette redéfinition se fait à l'intérieur de la définition de l'environnement `Ventry`, des doubles dièses doivent être utilisés pour différencier l'argument de la commande interne `\makeLabel` (`##1`) de l'argument de l'environnement `Ventry` (`#1`).

```
\newenvironment{Ventry}[1]{%
```

```

{\begin{list}}{\renewcommand{\makelabel}[1]{\textsf{##1 :}}\hfil}%
      \settowidth{\labelwidth}{\textsf{#1 :}}%
      \setlength{\leftmargin}{\labelwidth+\labelsep}}%
{\end{list}}

\begin{Ventry}{Correction des erreurs}
\item[Édition]
  Tout travail de composition d'un document commence par l'introduction
  du texte \a saisir dans l'ordinateur \a l'aide d'un éditeur.
\item[Compilation] En utilisant \LaTeX{}.
\item[Correction des erreurs]
  Les éventuelles erreurs sont corrigées ;
  on recompile ensuite le document.
\end{Ventry}

```

Édition : Tout travail de composition d'un document commence par l'introduction du texte à saisir dans l'ordinateur à l'aide d'un éditeur.

Compilation : En utilisant L^AT_EX.

Correction des erreurs : Les éventuelles erreurs sont corrigées ; on recompile ensuite le document.

Cependant, il n'est pas toujours acceptable, typographiquement parlant, que la largeur des termes des différentes listes sur une page varie. Pour remédier à une telle situation, on peut calculer la largeur du terme et, s'il est plus large que `\labelwidth`, une boîte vide est insérée de telle façon que la partie description commence sur une nouvelle ligne. Ceci correspond à la méthode usuelle pour lister les options des commandes dans les manuels Unix.

```

\newlength{\Mylen}
\newcommand{\Lentrylabel}[1]{%
  \settowidth{\Mylen}{\textsf{#1 :}}%
  \ifthenelse{\lengthtest{\Mylen > \labelwidth}}%
    {\parbox[b]{\labelwidth}{\makebox[0pt][l]{\textsf{#1 :}}\> \labelwidth
      {\textsf{#1 :}} < \labelwidth}
    \hfil\relax}
\newenvironment{Lentry}
  {\renewcommand{\entrylabel}{\Lentrylabel}\begin{entry}}
  {\end{entry}}

```

```

\begin{Lentry}
\item[Édition]
  Tout travail de composition d'un document commence par l'introduction

```

```

du texte \a saisir dans l'ordinateur \a l'aide d'un éditeur.
\item[Compilation] En utilisant \LaTeX{}.
\item[Correction des erreurs]
  Les éventuelles erreurs sont corrigées ;
  on recompile ensuite le document.
\end{Lentry}

```

Comme le montrent les dernières lignes de l'exemple, l'environnement `Lentry` est défini en fonction de l'environnement `entry`. La commande `\entrylabel`, qui génère l'étiquette, est remplacée par la commande `\Lentrylabel`. Cette dernière affecte d'abord la valeur de la largeur de l'étiquette à la dimension `\Mylen`. Puis elle compare cette dimension à `\labelwidth`. Si l'étiquette est plus étroite que `\labelwidth`, l'étiquette est composée sur la même ligne que sa description. Dans le cas contraire, l'étiquette (le terme de l'élément de la liste) est composée dans une boîte de largeur zéro avec son contenu dépassant du côté droit autant qu'il faut et forçant une nouvelle ligne, de telle façon que la partie description commence une ligne plus bas.

Édition :

Tout travail de composition d'un document commence par l'introduction du texte à saisir dans l'ordinateur à l'aide d'un éditeur.

Compilation :

En utilisant L^AT_EX.

Correction des erreurs :

Les éventuelles erreurs sont corrigées ; on recompile ensuite le document.

Une autre possibilité consiste à autoriser les étiquettes à occuper plusieurs lignes. Cette fois encore, on utilise l'environnement `entry`, mais la commande `\Mentrylabel` remplace `\entrylabel`. L'idée maîtresse est de diviser les longues étiquettes en plusieurs morceaux qui seront imprimés sur des lignes consécutives. Certaines précautions sont nécessaires pour garantir que le premier mot du paragraphe puisse être coupé. C'est pourquoi vous trouvez la commande `\hspace{0pt}` dans la définition de l'environnement. L'étiquette est composée dans une boîte d'une largeur adéquate égale à `\labelwidth`. La boîte est alignée à sa tête et le texte est justifié à gauche à l'intérieur de la boîte, qui est placée elle-même dans une boîte ayant une profondeur de 1 em et une hauteur nulle. De cette façon L^AT_EX

ne se rend pas compte que le contenu peut descendre en-dessous de la ligne de base de la première ligne.

```
\newcommand{\Mentrylabel}[1]{%
  \raisebox{0pt}[1em][0pt]{\makebox[\labelwidth][l]{%
    {\parbox[t]{\labelwidth}{\hspace{0pt}\textsf{#1 :}}}}}%
\newenvironment{Mentry}%
  {\begin{list}{}{\renewcommand{\makelabel}{\Mentrylabel}%
    \setlength{\labelwidth}{21mm}%
    \setlength{\leftmargin}{\labelwidth+\labelsep}}}%
  {\end{list}}
\begin{Mentry}
\item[Édition]
  Tout travail de composition d'un document commence par l'introduction
  du texte à saisir dans l'ordinateur à l'aide d'un éditeur.
\item[Compilation] En utilisant \LaTeX{}.
\item[Correction des erreurs]
  Les éventuelles erreurs sont corrigées ;
  on recompile ensuite le document.
\end{Mentry}
```

Édition : Tout travail de composition d'un document commence par l'introduction du texte à saisir dans l'ordinateur à l'aide d'un éditeur.

Compila- En utilisant L^AT_EX.
tion :

Correction Les éventuelles erreurs sont corrigées ; on recompile ensuite
des erreurs : le document.

Un environnement avec un compteur dont la valeur augmente d'une unité pour chaque élément de la liste peut se construire en introduisant une commande `\usecounter` dans la déclaration de l'environnement `list`. Une démonstration en est donnée dans l'environnement `Notes`, qui produit une séquence ordonnée de notes. Dans ce cas le premier argument de l'environnement `list` est utilisé pour le texte qui est généré automatiquement pour le terme de la liste.

Tout d'abord le compteur `notes` est déclaré, puis l'étiquette par défaut de l'environnement `Notes` est construite à partir du mot `NOTES` en petites capitales, suivi de la représentation en chiffres arabes de la valeur du compteur `notes` suivie d'un point.


```
\newcounter{notes}
\newenvironment{Notes}
  {\begin{list}{\textsc{Note} \arabic{notes}. }{%
    \usecounter{notes}%
    \setlength{\labelsep}{0pt}%
    \setlength{\leftmargin}{0pt}%
    \setlength{\labelwidth}{0pt}%
    \setlength{\listparindent}{0pt}}}%
  {\end{list}}

\begin{Notes}
\item Ceci est la première phrase de la première note.
      Et voici la deuxième phrase de la première note.
\item Ceci est la première phrase de la deuxième note.
      Et voici la deuxième phrase de la deuxième note.
\end{Notes}
```

NOTE 1. Ceci est la première phrase de la première note. Et voici la deuxième phrase de la première note.

NOTE 2. Ceci est la première phrase de la deuxième note. Et voici la deuxième phrase de la deuxième note.