

L'extension pour $\text{T}_\text{E}\text{X}/\text{L}\text{A}\text{T}_\text{E}\text{X}$

hlist

v 0.11

8 aout 2017

Christian TELLECHEA
unbonpetit@netc.fr

Cette petite extension sans prétention permet de créer des listes horizontale d'items ; s'ils s'étendent sur plus d'une ligne, ils seront alignés les uns au-dessous des autres, en colonnes de largeurs égales, un peu à la manière d'un tableau.

1 Comment ça marche

Au tout début, j'avais écrit ce code pour présenter des questions courtes *en ligne*, comme on peut le voir parfois dans les manuels scolaires, pour une économie de place verticale dans mes feuilles d'exercices ou de contrôles¹ :

```
Qu'est ce que \TeX ?
\begin{hlist}3
  \hitem Une marque \emph{distributeur} de vêtements que l'on trouve dans l'enseigne Carrefour;
  \hitem Une unité de mesure de la finesse de filaments;
  \hitem Un logiciel de composition créé par Donald \bsc{Knuth};
  \hitem Le nom \emph{de scène} de l'humoriste Jean-Christophe \bsc{Le Texier};
  \hitem Une membrane coupe-vent.
\end{hlist}
```

```
Qu'est ce que \TeX ?
```

1. Une marque <i>distributeur</i> de vêtements que l'on trouve dans l'enseigne Carrefour;	2. Une unité de mesure de la finesse de filaments;	3. Un logiciel de composition créé par Donald KNUTH;
4. Le nom <i>de scène</i> de l'humoriste Jean-Christophe LE TEXIER;	5. Une membrane coupe-vent.	

Le code initial dont je me suis servi pendant longtemps était bien plus basique que celui de cette extension; depuis, il s'est complexifié, mais les méthodes qu'il mettait en œuvre sont restées les mêmes. Or, j'avais programmé les listes horizontales en utilisant des *boîtes* de \TeX et non pas des *alignements*. Évidemment, il semble plus naturel de faire appel aux alignements de \TeX pour réaliser une telle tâche, mais leur mise en œuvre est plus délicate que celle des boîtes et j'ai été partisan du moindre effort (et je le suis toujours).

Récemment², alors que la finalisation du code avait déjà commencée, j'ai découvert que l'extension `Tasks` offrait *presque* les mêmes fonctionnalités que `hlist`. Comme *presque* n'est pas synonyme d'*exactement*, je me suis finalement décidé à publier cette extension. Il n'est bien entendu pas dans mon intention de dénigrer le travail de C. NIEDERBERGER et encore moins de considérer `hlist` comme meilleure que `Tasks`, mais au niveau des principales différences entre `Tasks` et `hlist`, on peut citer :

- `hlist` peut être utilisé avec \TeX (ε - \TeX en réalité) alors que `Tasks` requiert \LaTeX ;
- les listes de `hlist` peuvent être imbriquées tandis que celles de `Tasks` ne le peuvent pas;
- les changements de catcode (et donc le « verbatim ») sont possibles dans les listes de `hlist` et pas dans celles de `Tasks`;
- le code source de `hlist` est plus concis que celui de `Tasks` (écrit en $\LaTeX3$) et, contrairement à ce dernier, `hlist` ne fait appel à aucune extension tierce³;
- la documentation de `Tasks` est en anglais (son auteur est allemand); celle de `hlist` est en français. Cesser de docilement se plier à cette prétendue nécessité de fournir une documentation en anglais, fût-ce pour des extensions de \TeX/\LaTeX , compte *beaucoup* à mes yeux.

En revanche, puisque les deux extensions s'appuient sur des constructions de boîtes, `hlist` partage avec `Tasks` le principal défaut : une coupure de page ne peut pas avoir lieu au sein d'un item qui s'étendrait sur plusieurs lignes.

2 Syntaxe

L'environnement `hlist` peut être appelé de ces deux manières :

- façon \LaTeX : `\begin{hlist}[\langle param\grave{e}tres \rangle](\langle nombre \rangle)`
contenu de l'environnement
`\end{hlist}`
- façon \TeX : `\hlist[\langle param\grave{e}tres \rangle](\langle nombre \rangle)`
contenu de l'environnement
`\endhlist`

1. On ne trouve pas ce genre de questions dans mes contrôles!

2. <http://forum.mathematex.net/latex-f6/macros-a-parametres-indefini-t16733.html#p155215>

3. Vu les facilités prétendument offertes par $\LaTeX3$ et les extensions chargées par `Tasks`, je suis resté médusé face à la longueur du code de `Tasks`. Je ne m'explique pas une telle différence avec la taille de `hlist`, alors que `hlist` embarque son propre système de clés/valeurs, ses macros de développement, etc. Avant de me lancer dans une critique sans ménagement de $\LaTeX3$ et comme j'ai un fort soupçon que quelque chose m'échappe là-dedans, je serais très reconnaissant si quelqu'un, capable de comprendre ce qui se passe, pouvait me fournir une explication.

Les options, décrites plus loin, seront celles qui s'appliquent à cette liste et le $\langle nombre \rangle$ est le nombre de colonnes demandées.

L'environnement `hlist` doit commencer par un item, c'est-à-dire par la macro `\hitem`, éventuellement suivie des arguments optionnels suivants :

1. "*" signifiant que cet item doit commencer une nouvelle ligne;
2. ">" signifiant que cet item doit s'étendre sur toutes les colonnes restant disponibles sur la ligne en cours;
3. l'argument entre crochets " $\langle Label \rangle$ " spécifie ce que doit être la numérotation de cet item (voir la description de cette fonctionnalité plus loin);
4. l'argument entre parenthèses " $\langle n \rangle$ " où $\langle n \rangle$ est un entier, indique sur combien de colonnes doit d'étendre cet item.

Il faut noter que, s'ils sont présents tous les deux, l'argument optionnel "*" doit précéder ">". En revanche pour les deux derniers arguments optionnels, l'ordre n'a pas d'importance et donc " $\langle Label \rangle \langle n \rangle$ " et accepté tout comme " $\langle n \rangle \langle Label \rangle$ ".

<pre> \def\txtA{Après le pain, l'éducation est le premier besoin du peuple.} \def\txtB{Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes ?\par Cela doit tenir à l'éducation.} \hlist3 \hitem \txtA \hitem \txtB \hitem \txtA \hitem \txtA \hitem> \txtB \hitem*> \txtB \hitem(2) \txtB \hitem \txtA \endhlist </pre>		
<p>1. Après le pain, l'éducation est le premier besoin du peuple.</p>	<p>2. Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes? Cela doit tenir à l'éducation.</p>	<p>3. Après le pain, l'éducation est le premier besoin du peuple.</p>
<p>4. Après le pain, l'éducation est le premier besoin du peuple.</p>	<p>5. Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes? Cela doit tenir à l'éducation.</p>	
<p>6. Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes? Cela doit tenir à l'éducation.</p>		
<p>7. Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes? Cela doit tenir à l'éducation.</p>	<p>8. Après le pain, l'éducation est le premier besoin du peuple.</p>	

On peut voir à la page 6 cette même liste horizontale avec des cadres tracés autour des boites, ce qui peut rendre les choses plus claires pour cet exemple, volontairement complexe et peu réaliste.

3 Choix des paramètres

Nous avons déjà vu qu'écrire `\hlist[$\langle paramètres \rangle$]` permet de régler des $\langle paramètres \rangle$ spécifiques à la liste qui suit.

On peut également régler des $\langle paramètres \rangle$ pour toutes les `hlist` à venir avec

$$\backslash sethlist\{\langle paramètres \rangle\}$$

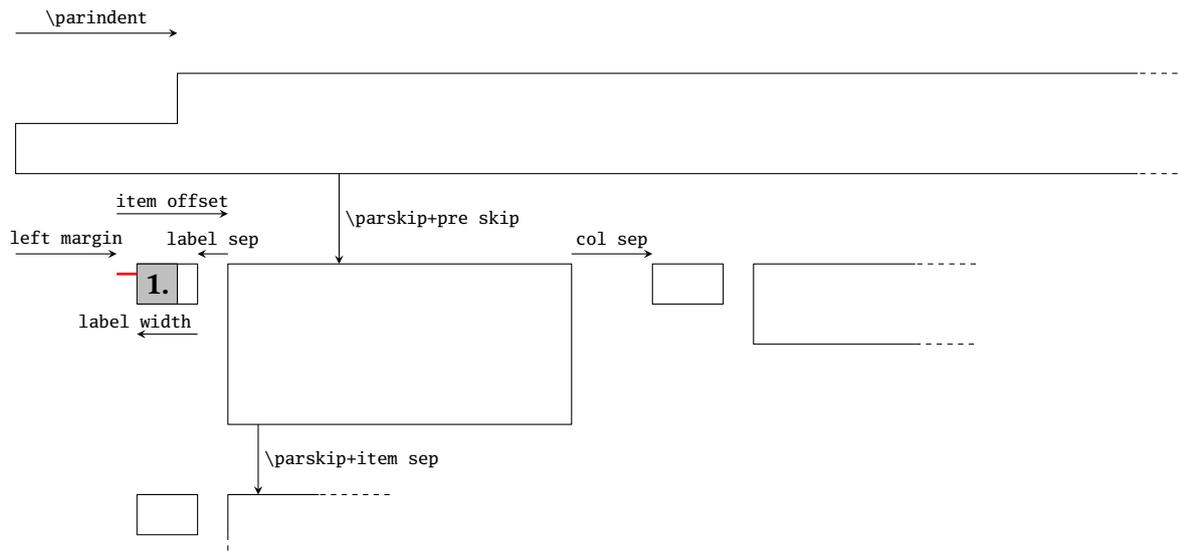
mais aussi régler des $\langle paramètres \rangle$ par défaut pour les `hlist` à venir avec

$$\backslash setdefault\hlist\{\langle paramètres \rangle\}$$

Chaque paramètre est de la forme $\langle clé \rangle = \langle valeur \rangle$ où la $\langle clé \rangle$ est le plus souvent constituée de deux mots (l'espace entre ces deux mots est nécessaire). Les espaces avant et après la $\langle clé \rangle$ et la $\langle valeur \rangle$ sont ignorés.

Lorsqu'une $\langle clé \rangle$ est seule, elle est comprise comme un booléen qui vaut true. Pour indiquer qu'un booléen est faux, il faut assigner la valeur false.

Les paramètres de *placement* sont résumés dans le schéma ci-dessous :



4 Placement de la liste de plus haut niveau

pre skip=*(ressort)* (Défaut : `\medskipamount`)
 Valeur du *(ressort)* vertical inséré avant la liste de plus haut niveau d'imbrication.

post skip=*(ressort)* (Défaut : `\medskipamount`)
 Valeur du *(ressort)* vertical inséré après la liste de plus haut niveau d'imbrication.

left margin=*(dimension)* (Défaut : `0pt`)
 Valeur du décalage vers la droite de la liste de plus haut niveau d'imbrication.

```

\parindent=2em
Ceci est un paragraphe s'étendant sur deux ligne et présentant
une liste de questions mises en forme à l'aide de l'extension \texttt{hlist}
et avec les paramètres par défaut :
\hlist3% paramètres par défaut
  \hitem Question 1      \hitem Question 2
  \hitem Une question 3
  \hitem Question 4      \hitem Question 5
\endhlist
Ceci est un paragraphe s'étendant sur deux ligne et présentant
une liste de questions mises en forme à l'aide de l'extension \texttt{hlist}
et avec des paramètres spécifiés par argument optionnel :
\hlist[pre skip = 0pt , post skip = 4pt , left margin = \parindent]3
  \hitem Question 1      \hitem Question 2
  \hitem Une question 3
  \hitem Question 4      \hitem Question 5
\endhlist
Suite du texte
    
```

Ceci est un paragraphe s'étendant sur deux ligne et présentant une liste de questions mises en forme à l'aide de l'extension `hlist` et avec les paramètres par défaut :

1. Question 1	2. Question 2	3. Une question 3
4. Question 4	5. Question 5	

Ceci est un paragraphe s'étendant sur deux ligne et présentant une liste de questions mises en forme à l'aide de l'extension `hlist` et avec des paramètres spécifiés par argument optionnel :

1. Question 1	2. Question 2	3. Une question 3
4. Question 4	5. Question 5	

Suite du texte

5 Label et texte des items

item offset=*(dimension)* (Défaut : 1.75em)

Cette *(dimension)* est la distance entre la frontière gauche de la boîte contenant le label et la frontière gauche de la boîte contenant le texte de l’item.

label sep=*(dimension)* (Défaut : 0.25em)

Espacement entre la boîte contenant le label et celle contenant le texte de l’item.

label width=*(dimension)* (Défaut : *)

C’est la dimension de la boîte accueillant le label. Le label lui-même (en gris sur la figure) peut être plus court que cette dimension auquel cas, il faut jouer avec la clé « `label align` » (voir plus bas) pour indiquer quel alignement on souhaite.

Si en revanche le label est *plus large* que la largeur « `label width` », alors un dépassement du type « `Overfull hbox` » aura lieu, sauf si on met la clé `autoindent` à la valeur `true` (voir plus bas).

Les trois clés « `item offset` », « `label sep` » et « `label width` » sont en interdépendance pour le placement du label et plusieurs cas peuvent se présenter :

- si l’une des 3 (et seulement *une seule* sur les trois) vaut « * », alors, les 2 autres sont calculées de telle sorte que l’égalité suivante soit vérifiée

$$\text{item offset} = \text{label sep} + \text{label width}$$

- si $\text{item offset} > \text{label sep} + \text{label width}$ comme c’est le cas sur la figure, alors la différence entre le membre de gauche et celui de droite (trait rouge sur la figure) est meublé avec un `\hfill` ;
- si $\text{item offset} < \text{label sep} + \text{label width}$, ce qui ne devrait pas se produire, un avertissement sera émis.

label align=*(lettre)* (Défaut : left)

Cette clé représente la consigne d’alignement du label à l’intérieur de sa boîte. Elle n’a de sens que lorsque le label occupe moins de place horizontale que « `label width` ». On peut demander une consigne `l`, `r` ou `c` selon que l’on souhaite un alignement à gauche, à droite ou centré.

Seule la première lettre *minuscule* de la valeur est prise en compte. La consigne `left` est donc équivalente à `l`.

label=*(code)* (Défaut : `\arabic {hlisti}.`)

Le *(code)* sera exécuté pour afficher le label. Pour afficher des labels en rapport avec le rang de l’item dans la liste, `hlist` met à disposition les macros suivantes :

- `\arabic{<nom>}` pour afficher un entier ;
- `\roman{<nom>}` et `\Roman{<nom>}` pour afficher des nombres romains, en minuscule ou majuscule ;
- `\alpha{<nom>}` et `\Alpha{<nom>}` pour afficher des lettres, minuscules ou majuscules.

L’argument de ces macros est le *nom* d’un compteur au format \LaTeX , c’est-à-dire que le registre de compteur qui se trouve derrière est `\c@<nom>`⁴.

`hlist` fournit les compteurs nommés « `hlisti` » pour les listes de plus haut niveau, « `hlistii` » pour les listes imbriquées de niveau 1, « `hlistiii` » pour celle de niveau 2, etc.

pre label=*(code)* (Défaut : `\bfseries`)

Contient le *(code)* qui sera exécutée avant d’afficher le label.

post label=*(code)* (Défaut : `{}`)

Contient le *(code)* qui sera exécutée après avoir affiché le label.

```
Exemple 1 :\hlist[label=\alpha{hlisti}]3
\hitem question 1 \hitem question 2 \hitem question 3 \hitem question 4
\endhlist
Exemple 2:\hlist[label=---]3
\hitem question 1 \hitem question 2 \hitem question 3 \hitem question 4
\endhlist
Exemple 3:\hlist[pre label=\begingroup\color{red}\itshape,label=\roman{hlisti}-,post label=\endgroup]3
\hitem question 1 \hitem question 2 \hitem question 3 \hitem question 4
\endhlist
```

4. Les compteurs de \LaTeX pour l’environnement `enumerate`, à savoir `enumi`, `enumii`, etc sont bien évidemment admis dans les arguments des macros ci-dessus. En règle générale, tout compteur de \LaTeX est également (section, equation, page, footnote ou autre).

Exemple 1 :		
a) question 1	b) question 2	c) question 3
d) question 4		
Exemple 2 :		
— question 1	— question 2	— question 3
— question 4		
Exemple 3 :		
<i>i-</i> question 1	<i>ii-</i> question 2	<i>iii-</i> question 3
<i>iv-</i> question 4		

Lorsque l'argument optionnel entre crochet de `\hitem` est vide, aucun label ne sera affiché et la valeur « item offset » sera nulle pour cet item. Cela peut s'avérer utile dans le cas de listes imbriquées :

<p>Exemple avec 2^{up} e label numéroté :</p> <pre>\hlist3 \hitem question 1 \hitem% label numéroté pour cet item ! \hlist[label=\alpha{hlistii}.,col sep=1em]3 \hitem XX \hitem YY \hitem ZZ \endhlist \hitem question 3 \hitem question 4 \hitem question 5 \endhlist</pre> <p>Même exemple avec 2^{up} e label forcé à vide :</p> <pre>\hlist3 \hitem question 1 \hitem[]% pas de label pour cet item ! \hlist[label=\arabic{hlisti}\alpha{hlistii}.]3 \hitem XX \hitem YY \hitem ZZ \endhlist \hitem question 3 \hitem question 4 \hitem question 5 \endhlist</pre>												
<p>Exemple avec 2^e label numéroté :</p> <table> <tr> <td>1. question 1</td> <td>2. a. XX b. YY c. ZZ</td> <td>3. question 3</td> </tr> <tr> <td>4. question 4</td> <td>5. question 5</td> <td></td> </tr> </table> <p>Même exemple avec 2^e label forcé à vide :</p> <table> <tr> <td>1. question 1</td> <td>2a. XX 2b. YY 2c. ZZ</td> <td>3. question 3</td> </tr> <tr> <td>4. question 4</td> <td>5. question 5</td> <td></td> </tr> </table>	1. question 1	2. a. XX b. YY c. ZZ	3. question 3	4. question 4	5. question 5		1. question 1	2a. XX 2b. YY 2c. ZZ	3. question 3	4. question 4	5. question 5	
1. question 1	2. a. XX b. YY c. ZZ	3. question 3										
4. question 4	5. question 5											
1. question 1	2a. XX 2b. YY 2c. ZZ	3. question 3										
4. question 4	5. question 5											

pre label=`(code)` (Défaut : `\bfseries`)
 Il s'agit du `(code)` qui est exécuté juste avant que le label soit affiché.

post label=`(code)` (Défaut : `{}`)
 Cette valeur contient le `(code)` qui est exécuté juste après l'affichage du label.

resume=`(booléen)` (Défaut : `false`)
 Lorsque cette clé est assignée à true, la numérotation de la liste n'est pas réinitialisée lorsque la macro `\hlist` est exécutée.

autoindent=`(booléen)` (Défaut : `false`)
 Lorsque la largeur du label dépasse la dimension « label width, nous avons vu qu'un avertissement du type `Overfull hbox` est produit. Lorsque la clé « autoindent » est mise à true, ce dépassement se fait en *indentant la première ligne du texte du label* de telle sorte que subsiste entre le label et le texte l'espace de largeur « label sep ». Il n'y a alors plus d'avertissement émis par \TeX .

Ceci fonctionne évidemment pour un schéma de label *global* passé par la clé « label=`{(code)}` », mais trouve surtout une utilité pour des labels *exceptionnels* spécifiés par `\hitem[Label]`.

```

\def\txt{et voici le texte de la question}
\hlist[item sep=5pt]3
  \hitem Question 1, \txt           \hitem Question 2, \txt           \hitem[Bonus] Question 3, \txt
  \hitem Question 4, \txt           \hitem Question 5, \txt
\endhlist
Texte intermédiaire...
\hlist[item sep=5pt,autoindent=true]3
  \hitem Question 1, \txt           \hitem Question 2, \txt           \hitem[Bonus] Question 3, \txt
  \hitem Question 4, \txt           \hitem Question 5, \txt
\endhlist

```

1. Question 1, et voici le texte de la question	2. Question 2, et voici le texte de la question	Bonus Question 3, et voici le texte de la question
4. Question 4, et voici le texte de la question	5. Question 5, et voici le texte de la question	

Texte intermédiaire...

1. Question 1, et voici le texte de la question	2. Question 2, et voici le texte de la question	Bonus Question 3, et voici le texte de la question
4. Question 4, et voici le texte de la question	5. Question 5, et voici le texte de la question	

pre item=(code) (Défaut : {})
 Le code exécuté juste avant le texte de l'item.

post item=(code) (Défaut : {})
 Le code exécuté juste après le texte de l'item.

show label=(booléen) (Défaut : true)
 Cette clé, lorsqu'elle est assignée à false, interdit l'affichage du label et donc, tout se passe comme si les dimensions item offset, label width et label sep valaient 0pt.

```

Exemple avec \texttt{show label} :
\hlist3
  \hitem Question 1           \hitem Question 2
  \hitem Une question 3
  \hitem Question 4           \hitem Question 5
\endhlist
Exemple sans \texttt{show label} :
\hlist[show label=false]3
  \hitem Question 1           \hitem Question 2
  \hitem Une question 3
  \hitem Question 4           \hitem Question 5
\endhlist

```

Exemple avec show label :

1. Question 1	2. Question 2	3. Une question 3
4. Question 4	5. Question 5	

Exemple sans show label :

Question 1	Question 2	Une question 3
Question 4	Question 5	

list parindent=(dimension) (Défaut : 0pt)
 Cette clé spécifie l'indentation des paragraphes de l'item, sauf pour le premier paragraphe de chaque item qui reste non indenté.

show frame=(booléen) (Défaut : false)
 Cette clé, lorsqu'elle est assignée à true, encadre les boîtes contenant le label et le texte de l'item.

```

\def\txtA{Après le pain, l'éducation est le premier besoin du peuple.}
\def\txtB{Comment se fait-il que les enfants étant si intelligents,
    la plupart des hommes soient bêtes ?\par Cela doit tenir à l'éducation.}
\hlist[show frame=true]3
  \hitem \txtA      \hitem \txtB      \hitem \txtA
  \hitem \txtA      \hitem> \txtB
  \hitem*> \txtB
  \hitem(2) \txtB      \hitem \txtA
\endhlist

```

1. Après le pain, l'éducation est le premier besoin du peuple.	2. Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes ? Cela doit tenir à l'éducation.	3. Après le pain, l'éducation est le premier besoin du peuple.
4. Après le pain, l'éducation est le premier besoin du peuple.	5. Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes ? Cela doit tenir à l'éducation.	
6. Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes ? Cela doit tenir à l'éducation.		
7. Comment se fait-il que les enfants étant si intelligents, la plupart des hommes soient bêtes ? Cela doit tenir à l'éducation.	8. Après le pain, l'éducation est le premier besoin du peuple.	

6 Le code

Contrairement à mes habitudes, le code est *abondamment* commenté, trop sans doute. J'espère que cela constituera une source d'information exploitable pour les utilisateurs de cette extension qui souhaitent y comprendre quelque chose ou modifier par eux-même une fonctionnalité.

Toute suggestion, remontée de bug, remarque, demande, ajout ou modification de fonctionnalité est bienvenue ; dans ce cas, j'invite les utilisateurs de `hlist` à m'envoyer un email.

Le code ci-dessous est l'exact verbatim du fichier `hlist.tex` :

```

1 % !TeX encoding = ISO-8859-1
2 % Ce fichier contient le code commenté de l'extension "hlist"
3 %
4 % IMPORTANT : pour que les commentaires s'affichent correctement,
5 % ouvrir ce fichier avec l'encodage ISO-8859-1
6 %
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8 %
9 \def\hlstname      {\hlist}
10 \def\hlstver      {0.11}
11 %
12 \def\hlstdate     {2017/08/08}
13 %
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 %
16 % -----
17 % This work may be distributed and/or modified under the
18 % conditions of the LaTeX Project Public License, either version 1.3
19 % of this license or (at your option) any later version.
20 % The latest version of this license is in
21 %
22 % %   http://www.latex-project.org/lppl.txt
23 %
24 % and version 1.3 or later is part of all distributions of LaTeX
25 % version 2005/12/01 or later.
26 % -----
27 % This work has the LPPL maintenance status 'maintained'.
28 %
29 % The Current Maintainer of this work is Christian Tellechea

```

```

30 % email: unbonpetit@netc.fr
31 %      Commentaires, suggestions et signalement de bugs bienvenus !
32 %      Comments, bug reports and suggestions are welcome.
33 % Package URL: https://www.ctan.org/pkg/hlist
34 % Copyright: Christian Tellechea 2017
35 % -----
36 % L'extension hlist est composée des 9 fichiers suivants :
37 %   - code           : hlist           (.tex et .sty)
38 %   - manuel en français : hlist-fr      (.tex et .pdf)
39 %   - fichiers de test  : hlist-latex    (.tex et .pdf)
40 %                       hlist-tex      (.tex et .pdf)
41 %   - fichier lisezmoi  : README
42 % -----
43
44 \begingroup
45 \ifdefined\skvname\let\hlsttmp\endgroup\else\def\hlsttmp{\endgroup\input simplekv.tex}\fi
46 \hlsttmp\relax
47 \expandafter\edef\csname hlst_restorecatcode\endcsname{\catcode'\noexpand\_=\the\catcode'\_ \relax}
48 \catcode'\_11
49
50 #####
51 ##### macros de développement #####
52 #####
53 % Cette macro est équivalente à 0 et sert notamment à stopper le développement
54 % de \romannumeral
55 \chardef\hlst_stop 0
56
57 % Définition du quark, notamment inséré à la fin d'une liste pour en reconnaître
58 % la fin et stopper la récursivité
59 \def\hlst_quark{\hlst_quark}
60
61 % Voici les macros habituelles de sélection d'arguments
62 \def\hlst_gobarg#1{}
63 \long\def\hlst_first#1#2{#1}
64 \long\def\hlst_second#1#2{#2}
65 \long\def\hlst_firsttonil#1#2\_nil{#1}
66
67 % Voici la macro pour 1-développer ou 2-développer le 2e argument (le 1er étant
68 % dépouillé des accolades)
69 %   \hlst_exparg{<a>}{<b>} devient <a>{<b>}
70 \long\def\hlst_exparg#1#2{\expandafter\hlst_exparg_i\expandafter{#2}{#1}}%
71 \long\def\hlst_exparg_i#1#2{#2{#1}}
72
73 % Et la macro pour 1-développer le 2e argument (le 1er et le 2e argument sont
74 % dépouillés des accolades)%
75 %   \hlst_expafater{<a>}{<b>} devient <a><{*b>
76 \long\def\hlst_expafater#1#2{\expandafter\hlst_expafater_i\expandafter{#2}{#1}}
77 \long\def\hlst_expafater_i#1#2{#2#1}
78
79 #####
80 ##### macros de test #####
81 #####
82 % Voici quelques macros à sélection d'arguments pour les tests
83 \def\hlst_ifnum#1{\ifnum#1\expandafter\hlst_first\else\expandafter\hlst_second\fi}
84 \def\hlst_ifcsname#1{\ifcsname#1\endcsname\expandafter\hlst_first\else\expandafter\hlst_second\fi}
85 \long\def\hlst_ifx#1{\ifx#1\expandafter\hlst_first\else\expandafter\hlst_second\fi}
86 \long\def\hlst_ifempty#1{\hlst_exparg\hlst_ifx{\expandafter\relax\detokenize{#1}\relax}}
87 \def\hlst_ifstar#1#2{\def\hlst_ifstar_i{\hlst_ifx{* \hlst_nxttok}{\hlst_first{#1}{#2}}\futurelet\hlst_nxttok\hlst_ifstar_i}
88
89 % Ici, une macro pour définir le token <espace>
90 \def\hlst_deftok#1#2{\let#1=#2\relax}\hlst_deftok\hlst_sptok{ }
91
92 % Et une macro pour tester le prochain token
93 \def\hlst_ifnxttok#1#2#3{\hlst_deftok\hlst_toktotest{#1}\def\_truecode{#2}\def\_falsecode{#3}\_
hlst_ifnxttok_a}

```

```

94 \def\hlst_ifnxttok_a{\futurelet\_futuralet\_futurtok\hlst_ifnxttok_b}
95 \def\hlst_ifnxttok_b{%
96   \hlst_ifx{\_futurtok\hlst_sp}
97   {\afterassignment\hlst_ifnxttok_a\let\_futuralet= }
98   {\hlst_ifx{\_futurtok\hlst_toktotest}\_truecode\_falsecode}%
99 }
100
101 #####
102 ##### macros de hlist #####
103 #####
104 % Voici les registres utilisés :
105 % 1) le compteur d'imbrication, incrémenté de 1 à chaque nouvelle liste
106 %   imbriquée :
107 \newcount\hlst_nest
108
109 % 2) Le compteur des \items sur chaque ligne :
110 \newcount\hlst_colcnt
111
112 % 3) La largeur des \hbox enfermant les <items> (un <item> est le <label> _et_
113 %   le <texte> de l'item) :
114 \newdimen\hlst_itemboxwidth
115
116 % 4) L'espace entre chaque <item> :
117 \newdimen\hlst_colsep
118
119 % 5) La largeur disponible pour loger une ligne d'<items> :
120 \newdimen\hlst_textwidth
121
122 % 6) La profondeur maximale des \vtop rencontrées jusqu'alors dans une ligne :
123 \newdimen\hlst_maxdepth
124
125 % 7) La profondeur du dernier élément de la \vtop la plus profonde de la ligne :
126 \newdimen\hlst_maxprevdepth
127
128 % 8) La boîte chapeau contenant l'<item> (<label> + <texte>) :
129 \newbox\hlst_currenthbox
130
131 % 9) La boîte contenant le <label> :
132 \newbox\hlst_labelbox
133
134 %10) La boîte contenant le <texte>
135 \newbox\hlst_textbox
136
137 %11) Un booléen à fins de débogage et une dimension de débogage
138 %   Il sera mis à faux après la phase de tests et supprimé dès
139 %   la version 0.2
140 \newif\ifhlstdebugmode\hlstdebugmodefalse
141 \newdimen\hlst_debugdim
142
143 % La macro qui écrit les avertissement dans le fichier log de façon équivalente
144 % à \wlog :
145 \def\hlst_warning#1{\immediate\write-1 {Package hlist Warning: #1}}
146
147 % Cette macro renvoie le 1er caractère détokénisé de la <valeur> obtenue par
148 % 2-développement de #1, de la forme \useKV[hlist]{<clé>}
149 \def\hlst_firstcarof#1{\expandafter\hlst_firsttonil\detokenize\expandafter\expandafter\expandafter{#1}\_nil}
150
151 % Cette macro évalue la dimension contenue dans la <valeur> d'une <clé>
152 % passée dans l'argument #1 et l'assigne à la <valeur> de cette <clé>
153 \def\hlst_evaldim#1{%
154 % On n'évalue que si la <valeur> ne commence pas par "*" qui est réservé
155 % \if\string*\hlst_firstcarof{\useKV[\hlstname]{#1}*}\else
156 % \expandafter\edef\csname hlst_[\hlstname]_\detokenize{#1}\endcsname{\the\dimexpr\useKV[\hlstname]{#1}\ }
157 %   relax}%
158 }

```

```

159
160 % Même principe pour les ressorts
161 \def\hlst_evalskip#1{\expandafter\edef\csname hlst_[\hlstname]_\detokenize{#1}\endcsname{\the\glueexpr\useKV[\hlstname]{#1}\relax}}
162
163 % Voici la macro par qui tout commence; elle peut être appelée par \hlist façon TeX
164 % ou par \begin{hlist} façon LaTeX.
165 % On teste d'abord si elle dispose d'un argument optionnel qui serait les
166 % <clé>=<valeur> applicables à cette liste
167 \def\hlist{\begingroup\hlst_ifnxttok[\hlst_i]{\hlst_i{}}
168 \def\hlst_i[#1]{%
169 % Si les <clé>=<valeur> ne sont pas vide, les définir dans le groupe semi simple
170 % qui vient d'être ouvert
171 \hlst_ifempty{#1}{\sethlist{#1}}%
172 % Fixer les dimensions et les ressorts
173 \hlst_evaldim{list parindent}\hlst_evaldim{item sep}\hlst_evaldim{left margin}\hlst_evaldim{col sep}%
174 \hlst_evaldim{item offset}\hlst_evaldim{label sep}\hlst_evalskip{pre skip}\hlst_evalskip{post skip}%
175 % Ensuite, lire un /entier/ et stocker dans \hlst_colcnt le nombre de colonnes voulu
176 \afterassignment\hlst_ii\hlst_colcnt=
177 }
178
179 \def\hlst_ii{%
180 % Incrémenter le compteur d'imbrications
181 \global\advance\hlst_nest 1
182 % Si le compteur de label correspondant à cette imbrication existe,
183 \hlst_ifcsname{c@hlist\romannumeral\hlst_nest}
184 % l'initialiser sauf si la <clé> "resume" vaut "true"
185 {\ifboolKV[\hlstname]{resume}{\csname c@hlist\romannumeral\hlst_nest\endcsname\hlst_stop}}
186 % S'il n'existe pas, le créer (attention, \newcount est \outer)!
187 {\csname newcount\expandafter\endcsname\csname c@hlist\romannumeral\hlst_nest\endcsname}%
188 \hlst_ifnum{\hlst_colcnt<1 }
189 % Envisager le cas où le nombre de colonnes demandé est < 1 et message d'erreur
190 {\errmessage{Invalid column number "\the\hlst_colcnt", 1 column used instead}\def\hlst_nbcoll{1}}
191 % La macro \hlst_nbcoll contient finalement le nombre de colonnes
192 {\edef\hlst_nbcoll{\the\hlst_colcnt}}%
193 % Si on doit afficher les labels
194 \ifboolKV[\hlstname]{show label}
195 % Calculer combien de dimensions explicites sont demandées par l'utilisateur
196 {\edef\hlst_implicitdim{\the\numexpr
197 \ifstring*\hlst_firstcarof{\useKV[\hlstname]{item offset}*}1\else0\fi+
198 \ifstring*\hlst_firstcarof{\useKV[\hlstname]{label width}*}1\else0\fi+
199 \ifstring*\hlst_firstcarof{\useKV[\hlstname]{label sep}*}1\else0\fi\relax
200 }%
201 % Si ce nombre est au moins 1
202 \ifnum\hlst_implicitdim>0
203 \hlst_ifnum{\hlst_implicitdim>1 }
204 % S'il est plus grand que 1, message d'erreur car parmi les 3 dimensions, _une_
205 % seule peut être implicite
206 {\errmessage{Too many implicit dimensions for labels, default settings used instead}%
207 \sethlist{item offset=1.5em,label sep=0.25em,label width=1.25em}%
208 }
209 % Si ce nombre est égal à 1, calculer la dimension implicite en fonction des 2
210 % autres dimensions explicites...
211 {\ifstring*\hlst_firstcarof{\useKV[\hlstname]{item offset}*}%
212 \edef\hlst_implicitdim{[\hlstname]{item offset=\the\dimexpr\useKV[\hlstname]{label sep}+\useKV[\hlstname]{label width}\relax}}%
213 \else\ifstring*\hlst_firstcarof{\useKV[\hlstname]{label sep}*}%
214 \edef\hlst_implicitdim{[\hlstname]{label sep=\the\dimexpr\useKV[\hlstname]{item offset}-\useKV[\hlstname]{label width}\relax}}%
215 \else\ifstring*\hlst_firstcarof{\useKV[\hlstname]{label width}*}%
216 \edef\hlst_implicitdim{[\hlstname]{label width=\the\dimexpr\useKV[\hlstname]{item offset}-\useKV[\hlstname]{label sep}\relax}}%
217 \fi\fi\fi
218 % ...et la sauvegarder dans le <trousseau> "hlist" via la <clé> correspondante
219 \expandafter\setKV\hlst_implicitdim
220 }%

```

```

221 \fi
222 % Vérifier que "item offset" >= label sep + label width et dans le cas
223 % contraire, avertir
224 \ifdim\useKV[\hlstname]{item offset}<\dimexpr\useKV[\hlstname]{label sep}+\useKV[\hlstname]{label width}
\relax
225 \hlst_warning{incompatibles dimensions: item offset (\the\dimexpr\useKV[\hlstname]{item offset}\relax) <
< label sep (\the\dimexpr\useKV[\hlstname]{label sep}\relax) + label width (\the\dimexpr\useKV[\hlstname]{label width}\relax)}%
226 \fi
227 }
228 {}%
229 % Si l'imbrication est 1, on est donc dans la hlist de plus haut niveau
230 \ifnum\hlst_nest=1 % si liste mère
231 % Régler la parindent à la <valeur> de la <clé> "left margin"
232 \parindent\dimexpr\useKV[\hlstname]{left margin}\relax
233 % et insérer le ressort "pre skip"
234 \ifhmode\par\fi\vskip\useKV[\hlstname]{pre skip}%
235 \else
236 % Sinon, pas d'indentation pour les listes imbriquées
237 \parindent0pt
238 \fi
239 % Sauvegarder la largeur actuelle horizontale
240 \hlst_textwidth=\dimexpr\ifdefined\linewidth\linewidth\else\hsize\fi-\parindent\relax
241 % Informations de débogage sur les dimensions des composants
242 \ifhlstdebugmode
243 \message{*** hlist mode debug : largeur = \the\hlst_textwidth}%
244 \hlst_debugdim=0pt \def\hlst_debuglineno{0}%
245 \edef\hlst_debuglineno{\number\numexpr\hlst_debuglineno+1}%
246 \def\debuginfoB{\space\space Ligne d'items no \hlst_debuglineno\space:^^J}%
247 \fi
248 \leavevmode
249 % Initialiser le compteur de colonnes \hlst_colcnt à 0
250 \hlst_colcnt0
251 % Sauvegarder dans \hlst_colsep la <valeur> de la <clé> "col sep"
252 \hlst_colsep\useKV[\hlstname]{col sep}\relax
253 % Au début de la liste : pas de multicolonne
254 \def\hlst_askedmulticol{1}%
255 % Calculer la largeur de chaque <item> (= <label> + <texte>) s'il est non multicol
256 \edef\hlst_colwidth{\the\dimexpr(\hlst_textwidth-\useKV[\hlstname]{col sep}*(\hlst_nbcot-1))/(\hlst_nbcot)}%
257 % Collecter dans une boîte tout ce qui se trouve jusqu'au premier \hitem
258 % (neutraliser l'espace, le tab et le retour charriot)
259 \setbox0\hbox\bgroup\catcode'\^^I 9 \catcode'\^^M 9 \catcode32 9
260 }
261
262 % Voici maintenant la macro qui ferme l'environnement. Elle aussi peut être
263 % à la TeX par \endhlist ou par \end{hlist} comme on le fait en LaTeX.
264 \def\endhlist{%
265 % En premier lieu, il faut exécuter la routine qui achève un \hitem (fermeture
266 % des boîtes, calcul de la \vtop la plus profonde et de son \prevdepth, etc)
267 \hlst_enditem
268 % Débogage
269 \ifhlstdebugmode
270 \message{\debuginfoB\space\space\space\space TOTAL = \the\hlst_debugdim/\the\hlst_textwidth^^J
271 \space\space\space\space Depassement = \number\numexpr\hlst_debugdim-\hlst_textwidth\relax sp^^J
272 *** Fin de la liste^^J^^J}%
273 \ifnum\numexpr\hlst_debugdim-\hlst_textwidth\relax>4 \errmessage{Depassement !}\fi
274 \fi
275 % Si la liste est non imbriquée, insérer le ressort "post skip"
276 \ifnum\hlst_nest=1 \vskip\useKV[\hlstname]{post skip}\fi
277 % Décrémenter le compteur d'imbrications
278 \global\advance\hlst_nest -1
279 % Fermer le groupe semi simple ouvert avec \hlist et manger les espaces
280 \endgroup
281 \ignorespaces
282 }

```

```

283
284 % Voici à présent des macros traduisant un <nombre> en un <label>.
285 % La première, qui est appelée par toutes les autres, converti le <nombre> en un
286 % nombre arabe.
287 \def\hlst_arabic#1{%
288 % Si le compteur (au modèle latex, c'est-à-dire "\c@<nom>") existe
289   \hlst_ifcsname{c@\detokenize{#1}}
290 % Donner le nombre arabe avec \number, sinon, message d'erreur
291   {\number\csname c@\detokenize{#1}\endcsname}
292   {\errmessage{Unknown counter named "\detokenize{#1}"}}%
293 }
294 % Ensuite, ces macros convertissent un <nombre> en nombre romain ou en lettre,
295 % avec choix de la casse
296 \def\hlst_roman#1{\romannumeral\hlst_arabic{#1}\relax}
297 \def\hlst_Roman#1{\uppercase\expandafter\expandafter\expandafter{\hlst_roman{#1}}}
298 \def\hlst_alpha#1{\ifcase\hlst_arabic{#1}\relax\or a\or b\or c\or d\or e\or f\or g\or h\or i\or j\or
299 k\or l\or m\or n\or o\or p\or q\or r\or s\or t\or u\or v\or w\or x\or
300 y\or z\else\errmessage{No letter for a value of \number\numexpr#1\relax}\fi}
301 \def\hlst_Alpha#1{\ifcase\hlst_arabic{#1}\relax\or A\or B\or C\or D\or E\or F\or G\or H\or I\or J\or
302 K\or L\or M\or N\or O\or P\or Q\or R\or S\or T\or U\or V\or W\or X\or
303 Y\or Z\else\errmessage{No letter for a value of \number\numexpr#1\relax}\fi}
304
305 % Voici maintenant la macro \hitem, qui marque le début d'un nouvel <item>
306 % Tout d'abord, on s'intéresse à ce qui suit, c'est-à-dire aux argument
307 % optionnels.
308 % On teste si elle est suivie d'une "*" (qui commande un retour à la ligne pour
309 % cet item) et on transmet le token 0 ou 1 selon l'issue du test
310 \def\hitem{\hlst_ifstar{\hlst_hitem_grabopts1}{\hlst_hitem_grabopts0}}
311
312 % Ensuite, on teste si elle est suivie de ">" (qui commande un multicol maximal
313 % pour cet item) et on transmet le token 0 ou 1 selon l'issue du test
314 \def\hlst_hitem_grabopts#1{\hlst_ifnxttok>{\hlst_first{\hlst_hitem_grabopts_i#11}}{\hlst_hitem_grabopts_i#
315 #10}}
316
317 % La macro \hlst_hitem_grabopts_i reçoit les 2 premiers arguments concernant la
318 % présence ou pas de "*" et de ">" ; ces 2 tokens seront transmis de proche en
319 % proche jusqu'à la macro \hitem_ii qui est la macro qui fait le boulot.
320 \def\hlst_hitem_grabopts_i#1#2{%
321 % On teste si une parenthèse (pour demander un multicol) est le prochain token
322   \hlst_ifnxttok(
323     {\hlst_hitem_grabopts_ii#1#2}
324 % Sinon, on teste si un crochet (pour forcer un label pour cet item) est le
325 % prochain token
326     {\hlst_ifnxttok[
327 % Si oui, aller lire ce qu'il contient
328       {\hlst_hitem_grabopts_iii#1#2}
329 % Si ni parenthèse ni crochet, transmettre "()[<quark>]" à \hitem_i
330       {\hitem_i#1#2()[\hlst_quark]}%
331     ]}%
332 }
333
334 % Ici, une parenthèse a été lue en premier, tester si la suite est un crochet
335 \def\hlst_hitem_grabopts_ii#1#2(#3){\hlst_ifnxttok[{\hitem_i#1#2(#3)}{\hitem_i#1#2(#3)[\hlst_quark]}}
336
337 % Ici, un crochet a été lu en premier, tester si la suite est une parenthèse
338 \def\hlst_hitem_grabopts_iii#1#2[#3]{\hlst_ifnxttok[{\hlst_hitem_grabopts_iv#1#2[#3]}{\hitem_i#1#2()[#3]}}
339
340 % Au cas où le crochet est lu en premier, remettre les argument optionnels dans
341 % le bon ordre : parenthèse _puis_ crochet
342 \def\hlst_hitem_grabopts_iv#1#2[#3](#4){\hitem_i#1#2(#4)[#3]}
343
344 % Cette macro mange tous les espaces _explicites_ qui se trouvent éventuellement
345 % après les arguments optionnels
346 % Pour résumer, \hitem peut être suivi de "*" _puis_ ">" (ordre imposé)
347 % Ensuite on peut trouver (<multicol>) et/ou [<label>] dans n'importe quel ordre

```

```

348 \def\hitem_i#1#2(#3)[#4]{%
349   \hlst_ifnxttok\hlst_sptok
350   {\def\__temp{\hitem_i#1#2(#3)[#4]}\expandafter\__temp\skv_gobspace}
351   {\hitem_ii#1#2(#3)[#4]}%
352 }
353
354 % Voici la /vraie/ macro qui va faire le boulot et ses arguments :
355 %   - #1 et #2 valent 0 ou 1 selon la présence de "*" et ">"
356 %   - #3 est le nombre de multicolonne demandé (si vide, comprendre 1)
357 %   - #4 est le label demandé (si <quark>, comprendre absence d'argument
358 %     optionnel)
359 \def\hitem_ii#1#2(#3)[#4]{%
360 % si ce n'est _pas_ le 1er de la liste
361   \hlst_ifnum{\hlst_colcnt>0 }
362 % Exécuter la routine de fin d'item
363   {\hlst_enditem}
364 % Sinon, fermer la boite ouverte par \hlist
365   {\egroup
366 % Si toutes ses dimensions ne sont pas nulles, message d'erreur
367   \ifnum0\ifnum\wd0=0 1\fi\ifnum\dp0=0 1\fi\ifnum\ht0=0 1\fi<111
368   \errmessage{\string\hitem\space expected}%
369   \fi
370 % comme il s'agit du 1er item de la liste, initialiser les profondeurs
371   \hlst_maxdepthOpt \hlst_maxprevdepthOpt
372   }%
373 % Incrémenter le compteur de label associé à cette imbrication
374   \expandafter\global\expandafter\advance\csname c@hlist\romannumeral\hlst_nest\endcsname 1
375 % Incrémenter le compteur de colonnes du nombre de colonnes sautées précédemment
376   \advance\hlst_colcnt\hlst_askedmulticol\relax
377 % Si "*", mettre le compteur de colonnes au maximum pour forcer retour à la
378 % ligne
379   \ifnum#1=1 \hlst_colcnt\numexpr\hlst_nbcoll+1\relax\fi
380   \ifnum\hlst_colcnt>1
381 % Si ce n'est pas le 1er item de la liste
382   \hlst_ifnum{\hlst_colcnt=\numexpr\hlst_nbcoll+1\relax}
383 % Si le nombre d'item sur la ligne a atteint son maximum,
384 % former le paragraphe, sauter l'espace verticale "item sep"
385   {\par\vskip\useKV[\hlstname]{item sep}\relax
386 % Débogage
387   \ifhlstdebugmode
388     \message{\debuginfoB\space\space\space\space TOTAL = \the\hlst_debugdim/\the\hlst_textwidth^^J
389     \space\space\space\space Depassement = \number\numexpr\hlst_debugdim-\hlst_textwidth\relax sp^^J}%
390     \ifnum\numexpr\hlst_debugdim-\hlst_textwidth\relax>\hlst_colcnt \errmessage{Depassement !}\fi% ↵
391     dépassement si plus que <nb colonnes>sp
392     \edef\hlst_debuglineno{\number\numexpr\hlst_debuglineno+1}%
393     \def\debuginfoB{\space\space Ligne d'items no \hlst_debuglineno\space:^^J}%
394     \hlst_debugdimOpt
395     \fi
396 % Tromper TeX : ne pas tenir compte des profondeurs des \vtop, mais utiliser la
397 % profondeur du dernier élément de la \vtop la plus profonde
398   \prevdepth\hlst_maxprevdepth
399 % Passer en mode horizontal, initialiser les variables pour commencer une
400 % nouvelle ligne d'items
401   \leavevmode
402   \hlst_maxdepthOpt \hlst_maxprevdepthOpt \hlst_colcnt=1
403   }
404 % S'il reste des items à loger sur la ligne, insérer espace inter-item
405   {\kern\hlst_colsep\relax
406 % Débogage
407   \ifhlstdebugmode
408     \edef\debuginfoB{\debuginfoB\space\space\space\space colsep (\the\hlst_colsep)^^J}%
409     \advance\hlst_debugdim\hlst_colsep
410     \fi
411   }%
412   \fi
413 % Calculer le nombre de multicolonne pour l'item courant

```

```

413 \edef\hlst_askedmulticol{\number\numexpr
414 \hlst_ifnum{#2=1 }
415 % Si ">", prendre le maximum disponible
416 { \hlst_nbcot+1-\hlst_colcnt}
417 % Sinon, prendre l'argument #3 (s'il est vide ce qui correspond à une absence
418 % d'argument optionnel, prendre 1)
419 { \hlst_ifempty{#3}{1}{(#3)}}%
420 \relax}%
421 \hlst_ifnum{\hlst_askedmulticol<1 }
422 % Si l'utilisateur est un joueur ou un abruti !
423 { \errmessage{Invalid multicol "(#3)", "(1)" used instead}\def\hlst_askedmulticol{1}}
424 % Si trop de multicol demandé
425 { \hlst_ifnum{\numexpr\hlst_colcnt+\hlst_askedmulticol-1\relax}>\hlst_nbcot\relax}
426 % Prendre le maximum et avertir
427 { \edef\hlst_askedmulticol{\number\numexpr\hlst_nbcot+1-\hlst_colcnt\relax}%
428 \hlst_warning{too much multicol (#3) asked, (\hlst_askedmulticol) used instead}}%
429 {}%
430 }%
431 % Calculer la largeur de la \hbox contenant l'<items>
432 \hlst_itemboxwidth\dimexpr\hlst_colwidth*\hlst_askedmulticol+\useKV[\hlstname]{col sep}*(\
433 \hlst_askedmulticol-1)\relax
434 % Définir la boîte contenant <label> + <texte>
435 \setbox\hlst_currenthbox\hbox to \hlst_itemboxwidth\bgroup
436 % Initialiser le débordement du <label>
437 \xdef\hlst_labeloverwidth{0pt}%
438 % Si on affiche les labels ET si #4 n'est pas vide (un label explicitement
439 % demandé "[ ]" doit être compris comme "ne pas afficher de label du tout pour
440 % cet item").
441 \hlst_ifnum{\ifboolKV[\hlstname]{show label}10\hlst_ifempty{#4}01=11 }%
442 % Ouvrir la boîte des labels
443 { \hbox to \useKV[\hlstname]{item offset}{%
444 % Pousser à droite au cas où "item offset">"label sep"+"label width"
445 \hfill
446 % Ouvrir la \hbox contenant le label et forcer sa dimension horizontale à la
447 % <valeur> "label width"
448 \hbox to \useKV[\hlstname]{label width}{%
449 % Trouver la première lettre de la <valeur> "label align", sachant qu'une
450 % consigne vide sera comprise comme un alignement "l"
451 \edef\hlst_align{\hlst_firstcarof{\useKV[\hlstname]{label align}l}}%
452 % Pousser à droite si cette lettre est "r" ou "c"
453 \if \string r\hlst_align\hfill
454 \else\if \string c\hlst_align\hfill
455 \fi\fi
456 % Localement, (re)définir les macros de formatage pour le label afin qu'elles
457 % deviennent accessibles à l'utilisateur
458 \let\arabic\hlst_arabic\let\roman\hlst_roman\let\Roman\hlst_Roman\let\alpha\hlst_alpha\let\Alpha\
459 \hlst_Alpha
460 % Mettre dans la boîte \hlst_labelbox tout ce qui concerne le label (les codes
461 % avant/après et le label formaté selon la <valeur> "label")
462 \setbox\hlst_labelbox\hbox{%
463 \useKV[\hlstname]{pre label}%
464 \hlst_ifx{\hlst_quark#4}{\useKV[\hlstname]{label}}{#4}%
465 \useKV[\hlstname]{post label}%
466 }%
467 % Mesurer le dépassement (signé) du label par rapport à sa dimension autorisée
468 % "label width"
469 \xdef\hlst_labeloverwidth{\the\dimexpr\wd\hlst_labelbox-\useKV[\hlstname]{label width}\relax}%
470 % Afficher la boîte contenant le label avec ou sans cadre
471 \ifboolKV[\hlstname]{show frame}
472 { \fboxrule.4pt \fboxsep-\fboxrule \fbox{\box\hlst_labelbox}}
473 { \box\hlst_labelbox}%
474 % Si on gère le dépassement de largeur
475 \ifboolKV[\hlstname]{autoindent}
476 { \ifdim\hlst_labeloverwidth>0pt \hss\fi}
477 {}%

```

```

477 % Pousser à gauche si l'alignement demandé est "l" ou "c"
478     \if      \string l\hlst_align\hfill
479     \else\if \string c\hlst_align\hfill
480     \fi\fi
481     }%
482 % Séparation horizontale entre boîte de <label> et boîte de <texte>
483     \kern\useKV[\hlstname]{label sep}\relax
484     }%
485 % Diminuer la largeur disponible de la largeur de la boîte de <label>, soit la
486 % <valeur> "item offset"
487     \advance\hlst_itemboxwidth-\useKV[\hlstname]{item offset}\relax
488     }
489     {}%
490 % Maintenant que la boîte du <label> est affichée (ou pas si c'est le choix de
491 % l'utilisateur), il faut passer à la boîte qui contient le <texte> du label.
492 % Bien entendu, on utilise des accolades _implicites_ puisque la fin de la boîte
493 % ne se situe pas dans cette macro.
494 % Cette boîte sera une \vtop...
495     \setbox\hlst_textbox\vtop\bgroup
496 % ...de largeur \hlst_itemboxwidth...
497     \hsize\hlst_itemboxwidth
498 % ...que l'on copie vers \linewidth pour les utilisateurs de LaTeX, afin que
499 % cette macro garde sa cohérence
500     \ifdefined\linewidth\linewidth\hlst_itemboxwidth\fi\relax
501 % Réglage du parindent à "list parindent"
502     \parindent\useKV[\hlstname]{list parindent}\relax
503 % Passage en mode H et pas d'indentation pour le 1er paragraphe si on
504 % affiche les labels
505     \ifboolKV[\hlstname]{show label}\noindent\indent
506 % Après être passé en mode horizontal, on insère l'indentation supplémentaire en
507 % cas de débordement de la largeur du <label>
508     \ifboolKV[\hlstname]{autoindent}
509     {\ifdim\hlst_labeloverwidth>0pt \kern\hlst_labeloverwidth\fi}
510     {}%
511 % Puis, on exécute le code inséré juste avant le <texte> de l'item
512     \useKV[\hlstname]{pre item}%
513 }
514 %
515 %
516 % | <texte> de l'item entre ces 2 macros |
517 % |_____|
518 %
519 % Une fois que le <texte> de l'item est /librement/ composé (librement, cela
520 % autorise les changements de catcode, le "verbatim", etc), la macro de fin du
521 % <texte> est exécutée.
522 \def\hlst_enditem{%
523 % D'abord, exécuter le code de fin d'item
524     \useKV[\hlstname]{post item}%
525 % Ensuite, former le paragraphe pour accéder à \prevdepth
526     \par
527 % et développer "\the\prevdepth" _avant_ de fermer la \vtop et la \hbox (le
528 % double \egroup) pour que cette dimension soit le texte de remplacement de
529 % \hlst_currentdepth
530     \hlst_exparg{%
531 % fermeture de la \vtop
532     \egroup
533     \ifboolKV[\hlstname]{show frame}{\fboxrule.4pt \fboxsep-.4pt \fbox{\box\hlst_textbox}}{\box\
hlst_textbox}%
534     \egroup
535     \def\hlst_currentdepth}{\the\prevdepth}%
536 % Si la profondeur de la \vtop qui vient d'être créé est supérieure à toutes
537 % celles rencontrées dans cette ligne,
538     \ifdim\dp\hlst_currenthbox>\hlst_maxdepth
539 % retenir la valeur de \prevdepth
540     \hlst_maxprevdepth\hlst_currentdepth\relax
541 % et mettre à jour la profondeur maximale des \vtop de la ligne en cours.

```

```

542 % Le but de cette manoeuvre est de trouver la profondeur du dernier élément de
543 % la plus profonde \vtop de la ligne en cours.
544 % Cette valeur de \prevdepth sera utilisée au prochain \par pour tromper
545 % TeX qui sinon, prendrait en compte la /grande/ profondeur des \vtop, menant à
546 % des lignes trop proches.
547   \hlst_maxdepth\dp\hlst_currenthbox
548   \fi
549 % Une fois tout ceci fait, afficher la boite contenant le <texte>
550   \box\hlst_currenthbox
551 % Débogage
552   \ifhlstdebugmode
553     \edef\debuginfoB{\debuginfoB\space\space\space\space item (\the\hlst_itemboxwidth)^^J}%
554     \advance\hlst_debugdim\hlst_itemboxwidth
555   \fi
556 }%
557
558 \hlst_restorecatcode
559 % Création, si nécessaire d'une macro \fbox, équivalente à celle de LaTeX
560 \unless\ifdefined\fbox
561 % Prendre les réglages par défaut identiques à ceux de LaTeX
562   \newdimen\fboxrule \newdimen\fboxsep \fboxrule=.4pt \fboxsep=3pt
563 % Pour les explications, voir codes 251 \a 254 de ce fichier :
564 %   http://progtex.fr/wp-content/uploads/2014/09/code.txt
565 % et pages 271 \a 274 de "Apprendre à programmer en TeX"
566   \def\fbox#1{%
567     \hbox{%
568       \vrule width\fboxrule
569       \vtop{%
570         \vbox{\hrule height\fboxrule \kern\fboxsep \hbox{\kern\fboxsep#1\kern\fboxsep}}%
571         \kern\fboxsep \hrule height\fboxrule
572       }\vrule width\fboxrule
573     }%
574   }
575 \fi
576
577 % Les réglages par défaut (modifiables avec \setdefaultthlist)
578 \setKVdefault[\hlstname]{
579   pre skip      = \medskipamount,% ressort vertical avant la liste mère
580   post skip     = \medskipamount,% ressort vertical après la liste mère
581   left margin   = Opt,% décalage à droite des premières boites de chaque ligne de la liste mère
582   col sep       = 2.5em,% espace de séparation entre colonnes
583   item offset   = 1.75em,% distance entre bord gauche du label et bord gauche de l'item
584   label sep     = 0.25em,% recul du label
585   label width   = *,% largeur du label automatique (=1.75-0.25 em) car "item ofset" = "label width" + "label
586   label align   = left,% consigne d'alignement
587   show label    = true,% afficher les numérotations et tenir compte de leur largeur
588   pre label     = \bf,% code exécuté au début de la boite contenant le label
589   label         = \arabic{hlisti}.,% code du label
590   post label    = {},% code exécuté après le label
591   item sep      = Opt,% espace vertical supplémentaire entre lignes d'items
592   list parindent= Opt,% indentation des paragraphes des items (sauf le 1er)
593   pre item      = {},% code exécuté avant le texte de l'item
594   post item     = {},% code exécuté après le texte de l'item
595   resume       = false,% pas de reprise de numérotation
596   autoindent    = false,% pas d'indentation auto en cas de label trop large
597   show frame    = false% ne pas afficher les cadres autour des boites <label> et <texte>
598 }
599
600 % La macro \inithlist remet toutes les <clés> à leur <valeur> par défaut
601 \expandafter\let\expandafter\inithlist\csname hlst_[\hlstname]\endcsname
602
603 % Macros permettant de modifier les <valeurs> des <clés> de hlist...
604 \def\sethlist#{\setKV[\hlstname]}
605
606 % ... ainsi que les <valeurs> par défaut

```

```
607 \def\setdefaultthlist#\setKVdefault[\hlstname]
608 \endinput
```

609
610 Versions :

Version	Date	Changements
0.1	01/05/2017	Première version
0.11	08/08/2017	Chargement de simpleKV