

Le package `tnspverb`

Code source disponible sur <https://github.com/typensee-latex/tnspverb.git>.

Version 0.1.0-beta développée et testée sur Mac OS X.

Christophe BAL

2020-10-25

Table des matières

I. Introduction	2
II. Packages utilisés	2
III. Écriture <code>pseudo verbatim</code>	2
IV. Historique	5
V. Toutes les fiches techniques	6
1. Écriture <code>pseudo verbatim</code>	6

I. Introduction

En complément de l'environnement `verbatim`, le package `tnspverb` propose `pseudoverb` un environnement pour écrire du contenu « pseudo verbatim » éventuellement dans un cadre avec un titre ou non. Par « pseudo verbatim », on entend du contenu verbatim où toutes les macros \LaTeX seront tout de même interprétées : le très gros du travail est fait par l'environnement `alltt` du package éponyme.

II. Packages utilisés

La roue ayant déjà été inventée, le package `tnspverb` utilise les packages suivants sans aucun scrupule.

- `alltt`
- `simplekv`
- `tcolorbox`

III. Écriture pseudo verbatim

Exemple 1 – Avec toutes les options

```
\begin{pseudoverb}[  
  title = Une sortie console,  
  frame,  
  center,  
  scale = .6  
]  
  Prix1 = 14 euros  
+ Prix2 = 30 euros  
-----  
  Total = 44 euros  
\end{pseudoverb}
```

Une sortie console

Prix1 = 14 euros
+ Prix2 = 30 euros

Total = 44 euros

Voici comment s'utilise chacune des options.

1. `title` permet de donner un titre **si un cadre est utilisé**. Un titre vide sera ignoré. Par défaut cette option est de valeur vide.
2. `frame` demande d'ajouter un cadre autour du contenu.
3. `center` sert à centrer le contenu.
4. `scale` permet de donner un coefficient d'échelle à appliquer à la largeur de la ligne. Par défaut cette option vaut 1.

Exemple 2 – Ajouter juste un cadre

```
\begin{pseudoverb}[  
  frame,  
  scale = .6  
]  
  Prix1 = 14 euros  
+ Prix2 = 30 euros  
-----  
  Total = 44 euros  
\end{pseudoverb}
```

Prix1 = 14 euros
+ Prix2 = 30 euros

Total = 44 euros

Exemple 3 – Sans option, un piège à éviter !

Voici un 1^{er} exemple montrant qu’il faut faire attention quand on ne veut pas indiquer d’option. Il semble naturel d’omettre `[]` avant le contenu mais ce qui suit met en lumière un petit problème quand on veut garder tous les espaces initiaux du contenu.

<pre>\begin{pseudoverb} Je suis en début de ligne !!! Moi je ne le suis pas. << --- DÉBUT DE LIGNE ICI. \end{pseudoverb}</pre>	<pre>Je suis en début de ligne !!! Moi je ne le suis pas. << --- DÉBUT DE LIGNE ICI.</pre>
--	--

Ce qui précède est tout à fait normal puisque \LaTeX ignore les espaces au début du contenu lorsqu’il commence par chercher un crochet ouvrant indiquant une option du point de vue de l’environnement. Une fois que \LaTeX tombe sur un caractère autre que `[`, l’option par défaut est utilisée mais malheureusement les espaces initiaux ont été mangés, ils sont perdus à tout jamais. Pour éviter le problème précédent, il suffit d’utiliser une option vide via `[]` comme ci-dessous.

<pre>\begin{pseudoverb}[] Je ne suis plus en début de ligne !!! Ouf il est bien comme moi. << --- DÉBUT DE LIGNE ICI. \end{pseudoverb}</pre>	<pre>Je ne suis plus en début de ligne !!! Ouf il est bien comme moi. << --- DÉBUT DE LIGNE ICI.</pre>
--	--

Le code suivant montre que \LaTeX peut voir une option là où l’on voudrait indiquer un contenu. Bien que l’option ne soit pas correcte, elle est silencieusement ignorée en coulisse. Ceci explique que ce qui est entre des crochets ci-dessous ne soit pas affiché.

<pre>\begin{pseudoverb} [Je ne suis pas du contenu même si je le souhaite] Il manque le début !!! \end{pseudoverb}</pre>	<pre>Il manque le début !!!</pre>
--	-----------------------------------

De nouveau la solution consiste à utiliser une option vide `[]` comme le montre l’exemple ci-après.

<pre>\begin{pseudoverb}[] [Je suis du contenu et non une option fausse] Il ne manque plus le début !!! \end{pseudoverb}</pre>	<pre>[Je suis du contenu et non une option fausse] Il ne manque plus le début !!!</pre>
---	---

Exemple 4 – Macros interprétées

Dans ce document, `\squaremacro` a été définie par `\newcommand\squaremacro{x^2}`. Le code ci-après montre que l’environnement `pseudoverb` interprète les macros¹.

1. Ceci permet de définir assez facilement des contenus indentés de type langage de programmation via la définition de macros de mise en forme des mots-clés.

<pre>\begin{pseudoverb} ===== Remarque importante ===== \squaremacro{} est interprétée mais pas cette formule x^2 ! \end{pseudoverb}</pre>	<pre>===== Remarque importante ===== x^2 est interprétée mais pas cette formule x^2 !</pre>
--	--

Exemple 5 – Du contenu sur plusieurs pages

Même avec un cadre un contenu pourra se trouver sur des pages successives.

<i>Sur deux pages</i>
<p>Bla, bla, bla</p> <p>Bla, bla, bla</p>
1
<p>Bla, bla, bla</p> <p>Bla, bla, bla</p>

IV. Historique

Nous ne donnons ici qu'un très bref historique récent ² de **tnspverb** à destination de l'utilisateur principalement. Tous les changements sont disponibles uniquement en anglais dans le dossier **change-log** : voir le code source de **tnspverb** sur **github**.

2020-10-25 Nouvelle version mineure **0.1.0-beta**.

- Pour l'environnement **peusodverb**, l'option **width** a été renommée **scale**.

2020-09-06 Première version **0.0.0-beta**.

2. On ne va pas au-delà de un an depuis la dernière version.

V. Toutes les fiches techniques

1. Écriture pseudo verbatim

```
\begin{pseudoverb} [#opt]  
  ...  
\end{pseudoverb}
```

— Option: on utilise un système clé/valeur.

1. `title` est le titre du cadre. Par défaut `title = {}`.
2. `frame` demande d'ajouter un cadre.
3. `center` sert à centrer le contenu.
4. `scale` donne le coefficient multiplicatif à appliquer à la largeur de la ligne. Par défaut `scale = 1`.