

# Le package `tnspverb`

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

Version 0.0.0-beta développée et testée sur Mac OS X.

Christophe BAL

2020-09-06

---

## Table des matières

I. Introduction	2
II. Packages utilisés	2
III. Écriture <code>pseudo verbatim</code>	2
IV. Historique	4
V. Toutes les fiches techniques	5
1. Écriture <code>pseudo verbatim</code>	5

---

# I. Introduction

En complément à l'environnement `verbatim` est proposé l'environnement `pseudoverb`, pour « pseudo verbatim », qui permet d'écrire, dans un cadre ou non, du contenu presque verbatim : ceci signifie que toutes les macros  $\text{\LaTeX}$  seront tout de même interprétées.

## II. Packages utilisés

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

- `alltt`
- `simplekv`

## III. Écriture pseudo verbatim

### Exemple 1 – Avec toutes les options

```
\begin{pseudoverb}[
  title = Une sortie console,
  frame,
  center,
  width = .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 clés.

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. `width` permet de donner un coefficient multiplicatif à appliquer à la largeur de la ligne. Par défaut cette option vaut 1.

### Exemple 2 – Juste un cadre en plus

```
\begin{pseudoverb}[
  frame,
  width = .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

Dans l'exemple suivant, il ne faut pas oublier d'utiliser `[]` avec l'environnement.

<pre>\begin{pseudoverb}[]   Prix1 = 14 euros + Prix2 = 30 euros -----   Total = 44 euros \end{pseudoverb}</pre>	<pre>Prix1 = 14 euros + Prix2 = 30 euros -----   Total = 44 euros</pre>
---	---

Voici ce qu'il se passe si l'on oublie `[]` avant le contenu <sup>1</sup>.

<pre>\begin{pseudoverb}   Je suis en début de ligne !   Aïe il ne devrait pas comme moi. \end{pseudoverb}</pre>	<pre>Je suis en début de ligne !   Aïe il ne devrait pas comme moi.</pre>
---	---

### Exemple 4 – Macros interprétées

Ci-après est utilisée `\squaremacro` qui a été définie par `\newcommand\squaremacro{$x^2$}`.

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

**Remarque.** Ceci permet de définir des contenus indentés de type langage de programmation assez facilement via la définition de macros de mise en forme de mots clé.

### Exemple 5 – Du contenu sur plusieurs pages

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

---

1. L<sup>A</sup>T<sub>E</sub>X ignore les espaces au début du contenu car il commence par chercher un crochet ouvrant indiquant une option du point de vue de l'environnement.

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

## IV. Historique

Nous ne donnons ici qu'un très bref historique récent <sup>2</sup> 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](#).

**2020-09-06** 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. `width` donne le coefficient multiplicatif à appliquer à la largeur de la ligne. Par défaut `width = 1`.