

Inclure des données calculées dans un document LaTeX

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 mai 2007

<https://www.bortzmeyer.org/donnees-calculees-dans-un-document-tex.html>

On a souvent besoin de mettre dans un document des données qui sont calculées, par exemple les unes par rapport aux autres (« trois serveurs à 1500 € font 4500 € ») soit extraites du monde extérieur, par exemple une base de données. Quelle est la façon la « meilleure » de le faire pour des documents LaTeX ?

Soit un document LaTeX avec un tableau comme :

```
\begin{tabular}{|l|r|r|r|}
\hline
\textbf{Poste} & \textbf{Coût unitaire} & \textbf{Quantité} & \textbf{Coût total} \\
\hline
\cline{1-4}
Café & 1,10 & 4 & 4,40 \\
...
```

Je voudrais que les données soient calculées automatiquement, c'est-à-dire que le total change si je change le coût unitaire ou bien la quantité.

LaTeX s'appuyant sur TeX, qui est un langage de programmation complet, la solution la plus propre est sans doute d'utiliser TeX. Oui, mais voilà, je n'ai jamais appris ce langage et je n'avais pas envie de m'y mettre. Notons aussi que ce langage permet des calculs comme celui ci-dessus mais pas, par exemple, des accès à une base de données.

Pour l'instant, la solution que j'utilise n'est pas très élégante mais relativement simple et très puissante puisqu'elle utilise un langage complet, Python. Elle repose sur un système de gabarits, Cheetah. Voici un exemple de code LaTeX+Cheetah (le `#echo` étant l'instruction Cheetah pour afficher le résultat d'une expression Python) :

```
#set cout_unitaire = 1.10
#set quantite = 4
\begin{tabular}{|l|r|r|r|}
\hline
\textbf{Poste} & \textbf{Coût unitaire} & \textbf{Quantité} & \textbf{Coût total} \\
\hline
\cline{1-4}
Café & $cout_unitaire & $quantite & \\
\hline
\hline
\end{tabular}
#echo $cout_unitaire * $quantite
\\
...
```

Ce code est traité ensuite par Cheetah :

```
% cheetah compile --iext tex_tmpl --oext tex_py propal.tex_tmpl
Compiling propal.tex_tmpl -> propal.tex_py
% python propal.tex_py > propal.tex
```

et on obtient un document LaTeX normal qu'on n'a plus qu'à traiter.