

\LaTeX & les tableaux, le package **tabularx**

Troisième partie : calcul automatique de la largeur des tableaux

Bertrand Masson

Les fiches de Bébert

26 août 2009

Un nouveau descripteur de colonne : X

Le package `tabularx` (`\usepackage{tabularx}`) introduit un nouveau type de colonne noté `X`. Celui-ci produit une colonne de type `\p{valeurUnité}` dont la valeur est automatiquement calculée par \LaTeX en fonction de la taille du tableau préalablement définie. Voici un exemple de tableau créé par `tabularx` :

Un truc	Première ligne longue, très longue, trop longue	une ligne un peu longue	Un petit quelque chose dans la dernière
Un autre truc	bof	rien	moins que rien

```
\begin{tabularx}{11cm}{|c|X|l|X|}
```

```
\hline
```

```
Un truc&Première ligne longue, très longue, trop longue
```

```
&une ligne un peu longue
```

```
&Un petit quelque chose dans la dernière\\
```

```
\hline
```

```
Un autre truc&ligne2&rien&moins que rien\\
```

```
\hline
```

```
\end{tabularx}
```

Un peu d'explication : la largeur du tableau

Elle est indiquée par `{11cm}` placé juste après le `\begin{tabularx}`. Pour l'exemple j'ai mis une mesure en centimètres mais dans la réalité je n'utilise que `{\textwidth}` ou `{\linewidth}` qui représente la longueur d'une ligne de texte (voir la fiche \LaTeX les unités & les longueurs). En effet j'utilise `tabularx` pour des tableaux très larges qui donc doivent prendre toute la largeur disponible, `{\textwidth}` ou `{\linewidth}` remplissent ce rôle.

Attention à l'indentation

Pour les fiches de Bébert j'ai supprimé l'indentation en ajoutant dans l'entête de mon source la commande `\setlength\parindent{0pt}` (voir la fiche \LaTeX les unités & les longueurs), donc je ne rencontre pas de problème. Mais si tu utilises dans ton document des retraits de paragraphe, ton tableau va être également indenté. La solution, ajouter `\noindent` qui surprime l'indentation pour le paragraphe qui suit l'instruction. Ton tableau ne sera plus indenté et les paragraphes qui suivent le resteront.

Un peu d'explication : la largeur du tableau

Pour comparer, voici les résultats obtenus avec une largeur de tableau de `{11cm}`, puis avec `{\textwidth}` :

Un truc	Première ligne longue, très longue, trop longue	une ligne un peu longue	Un petit quelque chose dans la dernière
Un autre truc	bof	rien	moins que rien

Un truc	Première ligne longue, très longue, trop longue	une ligne un peu longue	Un petit quelque chose dans la dernière
Un autre truc	bof	rien	moins que rien

Dans le deuxième exemple toute la largeur disponible est utilisée. Dans la suite de la fiche, je n'utiliserai plus que `{\textwidth}` pour les exemples de tableaux.

Un peu d'explication : les types de colonnes

L'instruction `{|c|X|l|X|}` fonctionne de la même manière que pour `tabular`, la seule différence avec `tabularx`, est la présence des `X`.

Comment ça marche ?

\LaTeX va attribuer aux colonnes de type `c`, `l` et `r` l'espace nécessaire définie par la donnée la plus grande de chaque colonne, puis l'espace restant sera attribué aux colonnes `X` à part égale. **Attention** toutes les colonnes de type `X` d'un même tableau auront la même largeur. La page suivante montre des exemples qui illustrent le fonctionnement de `tabularx`, en utilisant toujours `{|c|X|l|X|}` mais en variant le contenu des données des colonnes `c` et `l`.

Un peu d'explication : les types de colonnes

Un truc	Première ligne longue, très longue, trop longue	1259	Un petit quelque chose dans la dernière
Un autre truc	bof	2	moins que rien

Un truc beaucoup trop long, qui va mettre	Première ligne longue, très longue, trop longue	une ligne un peu longue	Un petit quelque chose dans la dernière
le bazar dans cette belle organisation	bof	rien	moins que rien

Un peu d'explication : les types de colonnes

Le dernier exemple montre qu'il faut réserver les colonnes `c`, `l` et `r` à celles dont le contenu est étroit et les `X` aux colonnes larges. Tu peux bien entendu n'employer que des colonnes `X` dans un même tableau. Le dernier exemple avec `{|X|X|X|X|}`.

Un truc beaucoup trop long, qui va mettre	Première longue, longue, longue	ligne très trop	une ligne un peu longue	Un petit quelque chose dans la dernière
le bazar dans cette belle organisation	bof		rien	moins que rien

tabularx & array

Tu peux utiliser avec **tabularx** les commandes de **array** (pas besoin de charger le package **array**). Tu peux utiliser par exemples les descripteurs `<\color{red}>` et `<\ttfamily>`. Le tableau précédent avec

```
{|>{\color{blue}\bfseries}X|>{\tiny}X|X<{\color{red}.}>{\ttfamily}X|}
```

Un truc beaucoup trop long, qui va mettre	Première ligne longue, très longue, trop longue	une ligne un peu longue.	Un petit quelque chose dans la dernière
le bazar dans cette belle organisation	bof	rien.	moins que rien

tabularx & les colonnes fusionnées

Toutes les commandes utilisées par **tabular** sont compatibles avec **tabularx**.
Donc aucun problème pour fusionner des colonnes avec `\multicolumn`

1	Première ligne longue, très longue, trop longue	1259	Un petit quelque chose dans la dernière
bof		9	moins que rien
3	re bof	100	encore moins que rien

```
\begin{tabularx}{\textwidth}{|c|X|r|X|}  
\hline  
1&Première ligne longue, très longue, trop longue&1259  
&Un petit quelque chose dans la dernière\\  
\hline  
\multicolumn{2}{|c|}{bof}&9&moins que rien\\  
\hline  
3&re bof&100&encore moins que rien\\  
\hline  
\end{tabularx}
```

Et pour des colonnes centrées verticalement c'est comment qu'on fait ?

Le package `tabularx` définit la colonne `X` par la commande suivante (voir la fiche \LaTeX créer ses commandes pour les explications) :

```
\newcommand{\tabularxcolumn}[1]{p{#1}}
```

Il te suffit de redéfinir la commande pour transformer le `p` en `m` ou `b` :

```
\renewcommand{\tabularxcolumn}[1]{m{#1}}
```

1	Première ligne longue, très longue, trop longue	1259	Un petit quelque chose dans la dernière	<pre>\renewcommand{\tabularxcolumn}[1]{m{#1}} \begin{tabularx}{\linewidth}{ c X r X } \hline 1&Première ligne longue, très longue, trop longue&1259 &Un petit quelque chose dans la dernière\\ \hline \multicolumn{2}{ c }{bof}&9&moins que rien\\ \hline 3&re bof&100&encore moins que rien\\ \hline \end{tabularx}</pre>
bof		9	moins que rien	
3	re bof	100	encore moins que rien	

Tu peux en profiter pour y ajouter des descripteurs :

```
\renewcommand{\tabularxcolumn}[1]{>{\tiny}m{#1}}
```

```
\renewcommand{\tabularxcolumn}[1]{>{\color{blue}}m{#1}}...
```

Je veux des colonnes X de tailles différentes, c'est comment qu'on fait ?

Pour cela il faut utiliser les descripteurs avec la commande `\hsize` de la façon suivante :

```
{|>{\setlength\hsize{0.5\hsize}}X|>{\setlength\hsize{1.5\hsize}}X|}
```

`\setlength` permet de modifier une longueur (voir la fiche \LaTeX les unités & les longueurs), `\hsize` correspond à la taille calculée par \LaTeX de la colonne X donc dans l'exemple ci dessus la taille de la première colonne est divisée par 2 (`0.5\hsize`) et la deuxième est augmentée de 50% (`1.5\hsize`).

Attention, voici la première contrainte de l'utilisation de `\hsize` : La somme de toute les colonnes X doit restée inchangée, donc la somme des coefficients multiplicateurs de `\hsize` doit être égale au nombre de colonnes X. Ici $0.5 + 1.5 = 2$. Si tu utilises 3 colonnes X tu peux avoir comme coefficients $0.5 + 2 + 0.5 = 3$ ou $1 + 1.8 + 0.2 = 3$...

La deuxième contrainte est que tu ne peux inclure une colonne X ainsi modifiée dans un `\multicolumn`.

Je veux des colonnes **X** de tailles différentes, c'est comment qu'on fait ?

Voici un exemple de l'utilisation de `\hsize` :

1	Première ligne longue, très longue, trop longue	1259	Un petit quelque chose dans la dernière
2	bof	9	moins que rien
3	re bof	100	encore moins que rien

```
\begin{tabularx}{\linewidth}{|c|>{\setlength\hsize{1.3\hsize}}X|r|>{\setlength\hsize{0.7\hsize}}X|}  
\hline  
1&Première ligne longue, très longue, trop longue&1259&Un petit quelque chose dans la dernière\\  
\hline  
2&bof&9&moins que rien\\  
\hline  
3&re bof&100&encore moins que rien\\  
\hline  
\end{tabularx}
```

Conclusion

Voilà encore une étape de franchie dans la construction des tableaux, et ce n'est pas fini. Une prochaine fiche décrira l'extension **tabulary** qui gère la largeur des colonnes en fonction de leur contenu.

À suivre. . .