

1.1 Introduction

Avec L^AT_EX, on peut produire des tableaux respectant une charte graphique favorisant la clarté des informations et contenant des fonctions d'automatisation et d'agencement permettant un rendu final irréprochable et « adaptatif ».

De prime abord, le principe de rédaction des tableaux en code peut paraître fastidieux et c'est pourquoi des macro-commandes ont été définies pour faciliter leur programmation. Néanmoins, une fois les instructions prise en main, il sera compliqué d'envisager rédiger des tableaux autrement que de la sorte.

1.2 Préceptes typographiques

Quand il s'agit de rédiger des tableaux, il existe des règles de « bonne pratique » que l'on retrouve appliquées dans la totalité des ouvrages comportant des tableaux de qualité :

- il ne doit y avoir *aucune* séparation verticale dans les tableaux, remplacée par des espaces.
- les blocs de texte d'une colonne ont une disposition *justifiée*.
- les éléments textuels seuls sont centrés horizontalement et verticalement (ou alignés verticalement en haut s'ils sont à côté d'une colonne justifiée).
- les nombres et grandeurs sont horizontalement et verticalement centrés (ou verticalement alignés en haut s'ils sont à côté d'une colonne justifiée).
- les devises sont horizontalement alignées à droite et verticalement centrées (ou verticalement alignées en haut si elles sont à côté d'une colonne justifiée).
- les unités sont insérées entre parenthèses dans le titre de la colonne.
- les titres de colonnes sont en gras et centrés horizontalement et verticalement.
- les titres de colonnes sont *toujours* au singulier.

Exemple 1.1: Tableau simple

Selon les critères listé ci-dessous, un tableau basique est produit avec l'environnement `\begin{tabular}<position>{<nombre et type de colonnes>}` et doit donc s'apparenter au code suivant :

Code

```
\begin{table}[H]
\caption{Un premier tableau simple avec \texttt{tabular}}
\begin{tabular}[t]{c p{5cm}}
\toprule
\thead{Titre colonne nombre (unité)} & \thead{Titre colonne texte} & \\
\midrule
1 & & Un premier paragraphe de \SI{5}{\centi\meter} de large & \\
2 & & Un deuxième paragraphe de \SI{5}{\centi\meter} de large & \\
3 & & Un troisième paragraphe de \SI{5}{\centi\meter} de large & \\
\bottomrule
\end{tabular}
\end{table}
```



Cela produira :

Titre colonne nombre (unité)	Titre colonne texte
1	Un premier paragraphe de 5cm de large
2	Un deuxième paragraphe de 5cm de large
3	Un troisième paragraphe de 5cm de large

TAB. 1.1 – Un premier tableau simple avec `tabular`

1.3 Structure de base

Si l'on se réfère au code de l'exemple 5.1 page 39, on fait donc appel à l'environnement `\begin{tabular}` pour produire un tableau sous sa forme la plus simple. Celui-ci présente une largeur conditionnée principalement par son contenu, c'est-à-dire que si le contenu des cellules est trop large, le tableau va déborder de l'espace de rédaction.

Si l'on décompose son code, on commence par considérer le tableau comme un élément *flottant* en l'insérant dans l'environnement `\begin{table}`. Dans cet environnement flottant est insérée la légende avec l'instruction `\caption{<légende>}` .

Ensuite est déclaré l'environnement `\begin{tabular}` qui produira le tableau. Cet environnement ne propose que le nombre et type de colonnes comme argument obligatoire, selon l'instruction suivante `\begin{tabular}[<position>]{<nombre et type de colonnes>}`. De base, il existe six types de colonnes¹ :

colonne alignée à gauche (left) : `l` ;

colonne alignée au centre (center) : `c` ;

colonne alignée à droite (right) : `r` ;

paragraphe aligné en haut avec largeur spécifiée (paragraph) : `p{<largeur>}` ;

paragraphe aligné au milieu avec largeur spécifiée (middle) : `m{<largeur>}` ;

paragraphe aligné en bas avec largeur spécifiée (bottom) : `b{<largeur>}` .

La largeur à préciser dans les trois derniers argument est absolue, avec les unités détaillées dans la ?? page ??.

Ces variables déterminent donc le *format* du contenu des cellules (texte, liste, grandeur physique...) ainsi que le *nombre* de colonnes.

Selon l'exemple 5.1 page 39, le bloc `{c p{5cm}}` définit donc un tableau à deux colonnes, la première `c` étant centrée horizontalement et alignée verticalement en haut et la seconde `p{5cm}` contenant des paragraphes de 5cm de large.

L'argument facultatif [`<position>`] aligne verticalement le tableau par rapport à d'autres éléments du texte, selon les variables suivantes :

haut de la page (top) : `t` ;

milieu de la page (middle) : `m` ;

bas de la page (bottom) : `b` .

1. D'autres types de colonnes ont été définis pour d'autres types de tableaux en ?? page ??



À l'intérieur de l'environnement `\begin{tabular}` est donc inséré le contenu du tableau, encadré par les instructions suivantes :

- `\toprule` : ligne épaisse pour encadrer le haut du tableau, sans retour à la ligne `\\` ;
- `\thead{<titre de la colonne>}` : insertion du titres des colonnes, avec un passage à la colonne suivante appelé avec le caractère `&` et un saut de ligne `\\` en fin de ligne ;
- `\midrule` : ligne fine pour séparer la ligne des titres de colonne du restant du tableau, sans retour à la ligne `\\` ;
- `<cellule 1> & <cellule 2>` : insertion du contenu des cellules, avec un passage à la colonne suivante appelé avec le caractère `&` et un saut de ligne `\\` en fin de ligne ;
- `\bottomrule` : ligne épaisse pour encadrer le bas du tableau, sans retour à la ligne `\\` .

Lorsqu'on code un tableau, il est important dans la mesure du possible de structurer le code à l'aide de tabulations, pour mettre en évidence les colonnes et s'y retrouver durant la programmation.

1.4 Tableau avec largeur relative

Dans la pratique, les tableaux produits avec l'environnement `\begin{tabular}` sont relativement rares et petits. Les rédacteurs feront plus souvent face à des cas de figures où les tableaux ont une largeur occupant tout l'espace de rédaction.

Pour plus de cohésion graphique, il convient donc de corrélérer automatiquement la largeur du tableau à la largeur des colonnes. Cela est rendu possible avec l'environnement `\begin{tabularx}`, qui introduit un nouveau format de colonne `X` à la largeur adaptative selon l'espace restant. Ce type de tableau reste cependant relativement concis et ne peut pas être réparti harmonieusement sur plus d'une page.

Exemple 1.2: Tableau à la largeur relative

Un tableau à la largeur relative est produit avec l'environnement `\begin{tabularx}{<largeur du tableau>[<position>]{<type de colonnes>}`. On remarquera le nouvel argument `{<largeur du tableau>}`, ainsi qu'un nouveau format de colonne `X` :

Code

```
\begin{table}[H]
\caption{Un deuxième tableau avec \texttt{tabularx}}
\begin{tabularx}{\linewidth}[t]{c X r}
\toprule
\thead{Titre colonne nombre (unité)} & \thead{Titre colonne
texte} & \thead{Prix (\EUR{}} \\
\midrule
1 & Un premier texte à l'agencement format \texttt{paragraph} et à la
largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée
\Verb{\linewidth} & 15 \\
2 & Un deuxième texte à l'agencement format \texttt{paragraph} et à la
largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée
\Verb{\linewidth} & 30 \\
3 & Un troisième texte à l'agencement format \texttt{paragraph} et à la
largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée
\Verb{\linewidth} & 45 \\
\bottomrule
\end{tabularx}
\end{table}
```

Cela produira :



Titre colonne nombre	Titre colonne texte	Devise (€)
1	Un premier texte à l'agencement format paragraph et à la largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée	15
2	Un deuxième texte à l'agencement format paragraph et à la largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée	30
3	Un troisième texte à l'agencement format paragraph et à la largeur automatiquement ajustée à l'espace restant sur la largeur spécifiée	45

TAB. 1.2 – Un deuxième tableau avec `tabularx`

La colonne X voit donc sa largeur s'adapter en fonction de son contenu et des celui des autres cellules pour mesurer précisément la variable `\linewidth` renseignée.

1.4.1 Tableau avec largeur relative



Temporary page!

L^AT_EX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.

If you rerun the document (without altering it) this surplus page will go away, because L^AT_EX now knows how many pages to expect for this document.