

© 2015 Université Laval



Cette création est mise à disposition selon le contrat **Attribution-Partage dans les mêmes conditions 4.0 International** de Creative Commons. En vertu de ce contrat, vous êtes libre de :

- ▶ **partager** — reproduire, distribuer et communiquer l'œuvre ;
- ▶ **remixer** — adapter l'œuvre ;
- ▶ utiliser cette œuvre à des fins commerciales.

Selon les conditions suivantes :



Attribution — Vous devez créditer l'œuvre, intégrer un lien vers le contrat et indiquer si des modifications ont été effectuées à l'œuvre. Vous devez indiquer ces informations par tous les moyens possibles, mais vous ne pouvez suggérer que l'Offrant vous soutient ou soutient la façon dont vous avez utilisé son œuvre.



Partage dans les mêmes conditions — Dans le cas où vous modifiez, transformez ou créez à partir du matériel composant l'œuvre originale, vous devez diffuser l'œuvre modifiée dans les même conditions, c'est à dire avec le même contrat avec lequel l'œuvre originale a été diffusée.

Notes de cours et exercices développés par Vincent Goulet, professeur titulaire, avec la contribution financière de la Bibliothèque de l'Université Laval.

Table des matières

1	Boîtes, tableaux et figures	1
1.1	Boîtes	1
1.2	De la conception de beaux tableaux	5
1.3	Tableaux	7
1.4	Éléments flottants	11
2	Document contenu dans plusieurs fichiers	13
2.1	Insertion de contenu avec la commande <code>\input</code>	13
2.2	Insertion de parties avec la commande <code>\include</code>	14
2.3	Exercices	16

1 Boîtes, tableaux et figures

Les tableaux et graphiques ne sont pas les éléments de texte les plus simples et rapides à créer avec \LaTeX . Les traitements de texte brillent, ici, avec leurs interfaces graphiques permettant de composer un tableau ou un graphique simple pièce par pièce avec la souris.

En revanche, pour ce type de contenu comme pour tout autre, \LaTeX fait exactement ce qu'on lui demande. À ce chapitre, les traitements de texte ne brillent plus ! Si vous avez déjà eu de la difficulté à contrôler les bordures d'un tableau, la hauteur des lignes ou la largeur des colonnes, vous comprenez ce à quoi nous faisons référence.

Avant de discuter de la création ou de l'insertion de tableaux, graphiques et images, il convient ...

1.1 Boîtes

Il arrive que l'on doive traiter de manière spéciale une aire rectangulaire de texte ; pour l'encadrer, la mettre en surbrillance ou la mettre en exergue, par exemple.

Avec les traitements de texte, on aura souvent recours aux tableaux à de telles fins. Or, les tableaux devraient être réservés pour disposer de l'information sous forme de grille. Pour disposer et mettre en forme du contenu se présentant sous forme rectangulaire, \LaTeX offre la solution plus générale des «boîtes».

Il n'est pas inutile de savoir que \TeX ne manipule que cela, des boîtes. Pour \TeX , chaque caractère, chaque lettre n'est qu'un rectangle d'une certaine largeur qui s'élève au-dessus de la ligne de base (les lignes d'une feuille lignée) et qui, parfois, se prolonge sous la ligne de base (pensons aux lettres p , y ou Q). Les commandes ci-dessous permettent simplement de créer d'autres boîtes dont le contrôle des dimensions et du contenu est laissé à l'utilisateur. Une fois créée, une boîte ne peut être scindée en parties.

Il existe trois sortes de boîtes en \LaTeX : les boîtes horizontales dont le contenu est disposé exclusivement côte à côte ; les boîtes verticales qui peuvent contenir plusieurs lignes de contenu ; les boîtes de réglure pour former des lignes pleines de largeur et hauteur quelconques.

1.1.1 Boîtes horizontales

Le plus simple concept de boîte dans \LaTeX est celui de boîte «horizontale», c'est-à-dire dont le contenu est disposé latéralement de gauche à droite¹. Le contenu est normalement du texte, mais conceptuellement ce pourrait être n'importe quoi, y compris d'autres boîtes.

Les commandes de base pour créer des boîtes horizontales :

```
\mbox{texte}
\fbox{texte}
```

Elles produisent une boîte de la largeur précise de *texte*. Avec la commande `\fbox`, le texte est au surplus encadré. En usage courant, la commande `\mbox` sert principalement à deux choses :

1. réunir en un bloc du texte que l'on ne veut pas voir scindé entre les lignes ou entre les pages ;
2. avec `\mbox{}`, laisser croire à \TeX que du contenu apparaît à un endroit sans toutefois qu'il n'occupe aucun espace.

Il existe également des versions plus générales des commandes `\mbox` et `\fbox`. Il s'agit de :

```
\makebox[largeur][pos]{texte}
\framebox[largeur][pos]{texte}
```

Les arguments optionnels *largeur* et *pos* déterminent respectivement la largeur de la boîte et la position du texte dans la boîte. Les valeurs possibles de *pos* sont : *l* pour du texte aligné à gauche, *r* pour du texte aligné à droite et *c* (la valeur par défaut) pour du texte centré. Ainsi, la commande

```
\framebox[3.5cm][l]{aligné à gauche}
```

produit aligné à gauche, alors que

```
\makebox[3.5cm]{centré}
```

produit centré .

1. D'où l'appellation *LR (left-right) box* en anglais.

1.1.2 Boîtes verticales

Les boîtes verticales se distinguent des boîtes horizontales par le fait qu'elles peuvent contenir plusieurs lignes de contenu empilées les unes au-dessus des autres. Lorsque le contenu en question est du texte, on obtient des paragraphes².

La commande de base pour créer une boîte verticale est

```
\parbox[pos]{largeur}{texte}
```

Ici, l'argument optionnel *pos* permet d'ajuster l'alignement vertical de la boîte avec la ligne de base : *b* ou *t* selon que l'on souhaite aligner, respectivement, le bas ou le haut de la boîte avec la ligne de base. Par défaut, la boîte est centrée avec la ligne de base. Cet argument n'a aucun effet si la boîte est le seul élément de contenu du paragraphe.

On remarquera que l'argument *largeur* est ici obligatoire. Autrement dit, on doit nécessairement définir la largeur des boîtes verticales, un peu comme il faut bien définir la largeur de la page pour le texte normal.

Les boîtes créées avec `\parbox` ne peuvent contenir de structures «complexes» comme des listes ou des tableaux. Parce que plus général, l'outil véritablement utile pour la création de boîtes verticales est l'environnement `minipage`. Cet environnement peut contenir à peu n'importe quel type de contenu. Comme son nom l'indique, c'est ni plus ni moins qu'une page miniature à l'intérieur de la page standard.

La syntaxe de l'environnement `minipage` est la suivante :

```
\begin{minipage}[pos]{largeur}
  texte
\end{minipage}
```

L'environnement `minipage` est souvent utilisé pour disposer des éléments de contenu de manière spécifique sur la page, notamment des tableaux ou des figures côte-à-côte ou en grille.

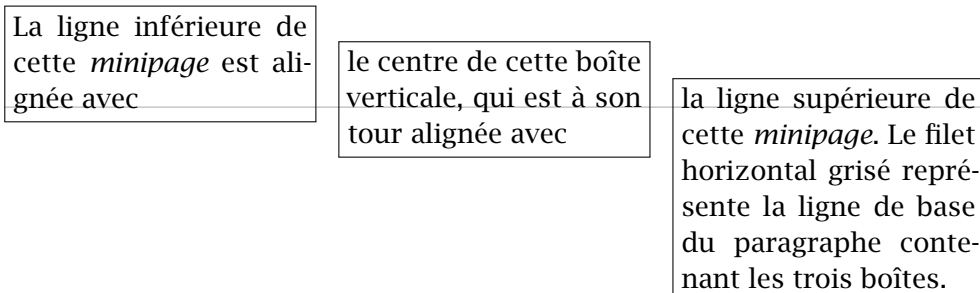
Voici un exemple d'utilisation des boîtes verticales. Le code suivant

```
\begin{minipage}[b]{0.3\textwidth}
  La ligne inférieure de cette \emph{minipage} [...]
\end{minipage}
\hfill
\parbox{0.3\textwidth}{le centre de cette boîte [...] }
\hfill
```

2. D'où l'appellation, cette fois, de *paragraph boxes* en anglais ou *parboxes* dans le jargon \LaTeX .

```
\begin{minipage}[t]{0.3\textwidth}
  la ligne supérieure de cette \emph{minipage}. [...]
\end{minipage}
```

produit :



La commande `\hfill` utilisée entre les boîtes ci-dessus indique à \LaTeX d'insérer de l'espace blanc entre les éléments de contenu de manière à remplir entièrement la ligne de texte. C'est une commande très utile pour disposer automatiquement des éléments à intervalles égaux sur la largeur du bloc de texte. Ainsi,

```
gauche \hfill droite
```

produit

gauche		droite
--------	--	--------

alors que

```
gauche \hfill centre \hfill droite.
```

produit

gauche	centre	droite.
--------	--------	---------

Remarque. Dans les exemples de boîtes ci-dessus, nous avons encadré celles-ci afin de les mettre en évidence. Le code pour créer les cadres n'apparaît pas dans les extraits de code source.

1.1.3 Boîtes de réglure

En imprimerie, une réglure est une ligne droite continue ou pointillée. Une ligne n'étant jamais rien d'autre qu'un rectangle plein, si mince fut-il, la réglure est le troisième type de boîte³ dans \LaTeX .

La commande

3. *Rule box*, en anglais


```
\rule[déplacement]{largeur}{hauteur}
```

crée une réglure de dimensions $largeur \times hauteur$. Par défaut, la réglure s'appuie sur la ligne de base. Le résultat de

```
\rule{2cm}{6pt}
```

est donc une ligne pleine de 2 cm de long et 6 points d'épais : .

L'argument optionnel *déplacement* permet de déplacer verticalement la réglure au-dessus ou au-dessous de la ligne de base selon que *déplacement* est positif ou négatif. Avec les deux commandes

```
\rule[3pt]{2cm}{6pt}
\rule[-3pt]{2cm}{6pt}
```

on crée les réglures  et , dans l'ordre.

Un usage intéressant de la réglure consiste à faire croire à \TeX qu'une ligne est plus haute qu'il n'y paraît en insérant dans celle-ci une réglure de largeur nulle. Par exemple, la distance entre la présente ligne et les autres du paragraphe est plus grande que la normale parce que nous y avons inséré une réglure invisible avec

```
\rule[-8pt]{0mm}{24pt}
```

Ce truc est particulièrement utile pour augmenter la hauteur des lignes dans un tableau ; voir la [section 1.3](#).

1.2 De la conception de beaux tableaux

On utilise les tableaux pour disposer de l'information sous forme de grille. Ainsi, le premier réflexe pour les mettre en forme consiste-t-il souvent à mettre en évidence cette grille par le biais de filets⁴ horizontaux et verticaux.

C'est une mauvaise idée, une pratique à éviter. Vraiment !

Comparer les deux tableaux ci-dessous. Le premier est mis en forme selon une approche classique supportée depuis toujours par \LaTeX : filets doubles en entête et en pied de tableau, filets simples entre chaque ligne et entre les colonnes.

4. Communément appelés «lignes» dans le langage courant ou «bordures» dans les logiciels de traitement de texte. Dans la documentation en anglais, on parle de *rules*.

i	v	b_i	$\lfloor v/b_i \rfloor$	$v \bmod b_i$	x_i
0	91 492	60	1 524	52	52
1	1 524	60	25	24	24
2	25	24	1	1	1
3	1	365	0	1	1

Le second tableau tire profit des fonctionnalités du paquetage **booktabs** et des recommandations de son auteur : les filets horizontaux sont d'épaisseur différente selon qu'ils sont situés dans l'entête et dans le pied du tableau ou entre les lignes, l'espace autour des filets horizontaux est plus grand et, surtout, il n'y a pas de filets verticaux.

i	v	b_i	$\lfloor v/b_i \rfloor$	$v \bmod b_i$	x_i
0	91 492	60	1 524	52	52
1	1 524	60	25	24	24
2	25	24	1	1	1
3	1	365	0	1	1

La seconde version n'est-elle pas la plus aérée et la plus facile à consulter ? Vous constatez que, contrairement à ce que l'on pourrait penser, les filets verticaux ne sont pas du tout requis pour bien délimiter les colonnes ?

Tel que mentionné ci-dessus, le paquetage **booktabs** ajoute des fonctionnalités à \LaTeX pour améliorer la qualité typographique des tableaux. Dans la documentation du paquetage, son auteur énonce quelques règles à suivre pour la mise en forme des tableaux :

1. ne *jamais* utiliser de filets verticaux. Si l'information du côté gauche du tableau semble si différente de celle du côté droit qu'un filet apparait vertical nécessaire, scinder simplement l'information dans deux tableaux ;
2. ne jamais utiliser de filets doubles ;
3. placer les unités dans le titre de la colonne plutôt qu'après chaque valeur dans le corps du tableau ;
4. toujours inscrire un chiffre du côté gauche du séparateur décimal : 0,1 et non ,1 (pratique plus répandue en anglais, où le séparateur décimal est le point) ;
5. ne pas utiliser un symbole pour représenter une valeur répétée (comme " ou —). Laisser un blanc ou répéter la valeur s'il subsiste une ambiguïté.

Nous recommandons évidemment de suivre ces règles et c'est pourquoi la présente documentation ainsi que les fichiers d'exemples font usage des commandes de **booktabs**.

Les fonctionnalités de **booktabs** sont intégrées à la classe `memoir` et par conséquent à `ulthese`. Il n'est donc pas nécessaire de charger le paquetage avec ces deux classes.

1.3 Tableaux

Peu importe l'outil utilisé, la création d'un tableau requiert de préciser à l'ordinateur le nombre de colonnes que contiendra le tableau, l'entête du tableau si nécessaire et le contenu des différentes cellules. Cette dernière étape nécessite à son tour une convention pour identifier les passages à la colonne suivante ainsi que le passage à la ligne suivante.

On crée des tableaux dans \LaTeX principalement avec les environnements `tabular`, `tabular*` et `tabularx` (ce dernier fourni par le paquetage **tabularx** ou par la classe `memoir`). La syntaxe de ces environnements est :

```
\begin{tabular}{format}          lignes \end{tabular}
\begin{tabular*}{largeur}{format} lignes \end{tabular*}
\begin{tabularx}{largeur}{format} lignes \end{tabularx}
```

La signification des arguments⁵ est la suivante. Nous ne traitons ici que les options les plus souvent employées. Pour une liste plus exhaustive, consulter la documentation de la classe `memoir` (chapitre 11) ou ? (section **Tableaux**).

largeur Largeur hors tout d'un tableau avec les environnements `tabular*` et `tabularx`. Autrement, avec l'environnement `tabular`, la largeur d'un tableau est déterminée automatiquement pour contenir tout le contenu du tableau, quitte à dépasser dans la marge de droite.

La largeur du tableau est généralement exprimée en fraction de la largeur du bloc de texte. Celle-ci est accessible avec la commande `\textwidth`. Par exemple, les déclarations suivantes définissent respectivement des tableaux occupant toute la largeur d'une page et 80 % de la largeur de la page :

```
\begin{tabular*}{\textwidth}{format}
```

```
\begin{tabularx}{0.8\textwidth}{format}
```

5. Nous avons omis un argument optionnel à peu près jamais utilisé servant à spécifier l'alignement vertical du tableau par rapport à la ligne de base externe.

L'environnement `tabular*` joue sur l'espace entre les colonnes pour parvenir à la largeur prescrite, alors que `tabularx` joue sur la largeur des colonnes (voir ci-dessous).

format Le format des colonnes et, par le fait même, le nombre de colonnes puisque l'argument doit compter un symbole pour chaque colonne du tableau. Les principaux symboles de mise en forme des colonnes sont :

- `l` contenu de la colonne aligné à gauche ;
- `r` contenu de la colonne aligné à droite ;
- `c` contenu de la colonne centré ;
- `p{lgr}` contenu de la colonne traité comme un paragraphe de texte de largeur *lgr* ;
- `X` [environnement `tabularx` seulement] colonne dont la largeur peut être ajustée pour obtenir un tableau de la largeur prescrite ; identique à `p` par ailleurs.

Par exemple, pour définir un tableau à trois colonnes dont le contenu de la première est aligné à gauche, celui de la seconde à droite et celui de la troisième en texte libre dans une cellule de 5 cm de largeur, on utiliserait :

```
\begin{tabular}{lrp{5cm}}
```

Avec la déclaration suivante, la largeur de la troisième colonne sera automatiquement adaptée pour que le tableau occupe toute la largeur de la page :

```
\begin{tabularx}{\textwidth}{lrX}
```

Les symboles `|` et `||` dans *format* servent à insérer des filets verticaux simples et doubles entre les colonnes, mais nous avons vu à la [section 1.2](#) que c'est une pratique à proscrire.

lignes Le contenu des cellules du tableau. Les entrées des cellules sont séparées par le symbole `&` et les lignes par `\\`. Une cellule peut être vide.

Les lignes de contenu peuvent également contenir certaines commandes spéciales pour contrôler la mise en forme. La commande ci-dessous permet de fusionner des cellules :

`\multicolumn{num}{fmt}{texte}` fusionne les *num* cellules suivantes en une seule de format *fmt* et contenant le texte *texte*.

Cette commande ne peut apparaître qu’au début d’une ligne ou après un symbole de changement de colonne &.

La commande est souvent utilisée avec $num = 1$ pour changer le format d’une cellule, par exemple pour centrer le titre d’une colonne autrement alignée à gauche ou à droite.

Les commandes suivantes⁶ servent à insérer des filets horizontaux dans un tableau :

`\toprule` insère un filet horizontal épais suivi d’un espace vertical au début d’un tableau ;

`\midrule` insère un filet horizontal mince précédé et suivi d’un espace vertical entre deux lignes ;

`\cmidrule{n-m}` insère un filet horizontal comme `\midrule`, mais seulement de la gauche de la colonne n à la droite de la colonne m ;

`\bottomrule` insère un filet horizontal épais précédé d’un espace vertical à la fin d’un tableau.

Ces commandes doivent toutes obligatoirement être précédées d’une fin de ligne `\\`, sauf évidemment `\toprule`.

Un exemple simple de lignes de contenu serait :

```
\toprule
Produit & Quantité & Prix unitaire (\$) & Prix (\$) \\
\midrule
Vis à bois      & 2 & 9,90 & 19,80 \\
Clous vrillés & 5 & 4,35 & 21,75 \\
\midrule
TOTAL          & 7 &      & 41,55 \\
\bottomrule
```

La hauteur des lignes d’un tableau est déterminée automatiquement en fonction de contenu de celles-ci.

Exemple 1.1. On reprend le contenu ci-dessus pour en faire un tableau d’une largeur ajustée automatiquement au contenu. La première colonne est alignée à gauche et toutes les autres à droite.

6. Ce sont les commandes de **booktabs** et memoir auxquelles nous faisons référence à la [section 1.2](#).

```

\begin{tabular}{lrrr}
\toprule
Produit & Quantité & Prix unitaire (\$) & Prix (\$) \\
\midrule
Vis à bois & 2 & 9,90 & 19,80 \\
Clous vrillés & 5 & 4,35 & 21,75 \\
\midrule
TOTAL & 7 & & 41,55 \\
\bottomrule
\end{tabular}

```

Produit	Quantité	Prix unitaire (\$)	Prix (\$)
Vis à bois	2	9,90	19,80
Clous vrillés	5	4,35	21,75
TOTAL	7		41,55

Avec quelques modifications, le tableau occupe maintenant toute la largeur de la page, la largeur de la première colonne étant ajustée pour combler l'espace nécessaire. De plus, on modifie l'entête de la première colonne avec la commande `\multicolumn` afin de centrer le titre. Enfin, on augmente la hauteur de l'entête à l'aide d'une réglure invisible ([sous-section 1.1.3](#)).

```

\begin{tabularx}{\textwidth}{Xrrr}
\toprule
\multicolumn{1}{c}{Produit} & & & \\
\rule[-8pt]{0mm}{24pt} Quantité & & & \\
Prix unitaire (\$) & Prix (\$) & & \\
\midrule
Vis à bois & 2 & 9,90 & 19,80 \\
Clous vrillés & 5 & 4,35 & 21,75 \\
\midrule
TOTAL & 7 & & 41,55 \\
\bottomrule
\end{tabularx}

```

Produit	Quantité	Prix unitaire (\$)	Prix (\$)
Vis à bois	2	9,90	19,80
Clous vrillés	5	4,35	21,75
TOTAL	7		41,55

1.4 Éléments flottants

Les tableaux et figures sont des éléments de contenu qui occupent souvent beaucoup d'espace vertical dans la page. S'il ne reste plus assez de place pour afficher un tel élément sur une page, T_EX devra le déplacer au début de page suivante et cela risque de produire une page inesthétique car insuffisamment remplie⁷.

Pour éviter cela, il serait souhaitable que T_EX puisse insérer un tableau à l'endroit indiqué dans le code source s'il y a suffisamment d'espace sur la page pour l'accueillir

De plus, tout tableau et toute figure dans un document soigné devrait comporter un titre/légende/description ainsi qu'un numéro afin de pouvoir y faire référence dans le texte («comme l'illustre la figure 3...»).

fonctionnalité très élaborée de LaTeX

éléments flottants conseils de booktabs tableaux figures (insertion de graphiques (Sweave), mention de picture)

LATEX offre à ses valeureux utilisateurs la possibilité d'utiliser des environnements flottants. Ces environnements ont la particularité de rendre « flottants » leur contenu. C'est-à-dire que LATEX choisit à partir d'un algorithme qui tient compte d'un certain nombre de paramètres, la position de l'environnement dans le document.

7. *Underful \vbox* dans le jargon de T_EX.

2 Document contenu dans plusieurs fichiers

Un document \LaTeX comporte toujours un préambule suivi du corps du texte. Lorsque ceux-ci sont relativement courts (peu de commandes spéciales et moins d'une vingtaine de pages de texte), il demeure assez simple et convivial d'en faire l'édition dans un seul fichier à l'aide de son éditeur de texte favori.

Cependant, si le préambule devient long et complexe ou, surtout, lorsque l'ampleur du document augmente jusqu'à compter un grand nombre de pages sur plusieurs chapitres, il convient de répartir les divers éléments du document dans des fichiers séparés.

La segmentation en plusieurs fichiers rend l'édition du texte plus simple et plus efficace. De plus, elle peut significativement accélérer, durant la phase de rédaction, la compilation des documents très longs ou comportant plusieurs images.

2.1 Insertion de contenu avec la commande `\input`

La commande `\input` permet d'insérer le contenu d'un autre fichier dans un document \LaTeX .

- La syntaxe de la commande est

```
\input{fichier}
```

où le nom du fichier à insérer est *fichier.tex*. On laisse donc tomber l'extension *.tex* qui est implicite.

- Le contenu du fichier est inséré tel quel dans le document, comme s'il avait été tapé dans le fichier qui contient l'appel à `\input`.
- Le procédé est surtout utile pour sauvegarder séparément des bouts de code qui pourraient nuire à l'édition du texte (figures, longs tableaux) ou

qui sont communs entre plusieurs documents (licence d'utilisation, auteur et affiliation).

- La commande peut aussi être utilisée dans le préambule pour charger une partie ou l'ensemble de celui-ci. Cela permet de composer un même préambule pour plusieurs documents. Il suffit alors de faire d'éventuelles modifications à un seul endroit pour les voir prendre effet dans tous les documents.

2.2 Insertion de parties avec la commande `\include`

Extrait de la documentation de la classe `ulthese` :

«Il est recommandé de segmenter tout document d'une certaine ampleur dans des fichiers `.tex` distincts pour chaque partie — habituellement un fichier par chapitre. Le document complet est composé à l'aide d'un fichier maître qui contient le préambule \LaTeX et un ensemble de commandes `\include` pour réunir les parties dans un tout.»

Comme `\input`, la commande `\include` insère le contenu d'un autre fichier dans un document \LaTeX . Son effet est cependant différent et c'est son utilisation qui permet d'accélérer la compilation d'un long document.

- L'insertion d'un fichier avec `\include` débute toujours une nouvelle page. On utilisera donc `\include` principalement pour insérer des chapitres entiers plutôt que pour des portions de texte.
- Un fichier inséré avec `\include` peut contenir des appels à `\input`, mais pas à `\include`.
- La syntaxe de la commande `\include` est

```
\include{fichier}
```

où le nom du fichier à insérer est *fichier.tex*. Ici aussi on laisse tomber l'extension `.tex` qui est implicite.

- La structure type d'un fichier maître est la suivante :

```
\documentclass{ulthese}
[...]\begin{document}

\frontmatter
```

```
\include{introduction}
\tableofcontents*

\mainmatter
\include{historique}      % premier chapitre
\include{modele}         % deuxième chapitre
[...]

\end{document}
```

Le principal avantage de `\include` par rapport à `\input` réside dans le fait que \LaTeX peut préserver entre les compilations les informations telles que les numéros de pages, de sections ou d'équations, ainsi que les références. Cela permet, par exemple, de compiler le texte d'un seul chapitre — plutôt que le document entier — et de néanmoins obtenir du chapitre une image représentative. Procéder ainsi accélère significativement la compilation des documents longs ou complexes.

- La commande `\includeonly`, que l'on utilise exclusivement dans le préambule, sert à spécifier le ou les fichiers à compiler tout en préservant la numérotation et les références. Sa syntaxe est

```
\includeonly{liste_fichiers}
```

où *liste_fichiers* contient les noms des fichiers à inclure dans la compilation, séparés par des virgules et sans l'extension `.tex`.

- Lors de l'utilisation de la commande `\includeonly`, toute la numérotation dans les fichiers *liste_fichiers* suivra celle établie lors de la compilation précédente.
- Si l'édition des fichiers de *liste_fichiers* cause des changements dans la numérotation et les références dans les autres parties du document, une nouvelle compilation de l'ensemble ou d'une partie de celui-ci s'avérera nécessaire.

Supposons qu'un document est composé des fichiers `chapitre1.tex`, `chapitre2.tex`, `chapitre3.tex` et que les chapitres débutent respectivement aux pages 1, 23 et 41.

- Si seul le fichier `chapitre2.tex` est inclus dans la compilation, le numéro du chapitre sera toujours 2 et le folio de la première page sera toujours 23, même si les 22 pages précédentes ne se trouvent pas dans le document.

- Si l'on modifie le fichier `chapitre2` de telle sorte que le chapitre se termine maintenant à la page 46, il faudra recompiler le document avec au moins les fichiers `chapitre2` et `chapitre3` pour que les pages du chapitre 3 soient renumérotées à partir de 47.

L'exercice 2.1 illustre mieux le cycle typique d'utilisation des commandes `\include` et `\includeonly`.

2.3 Exercices

- 2.1 Cet exercice fait appel au fichier maître `exercice_include.tex` et à plusieurs fichiers auxiliaires. Schématiquement, le document est composé ainsi :

```
exercice_include.tex
├─ \input pagetitre.tex
├─ \include presentation.tex
│   └─ \includegraphics console-screenshot.pdf
└─ \include emacs.tex
```

La commande `\includegraphics` permet d'insérer une image en divers formats (PDF, JPEG, PNG, ...) dans un document \LaTeX . Elle provient du paquetage **graphicx**.

- Étudier le code source du fichier maître `exercice_include.tex`, puis le compiler deux à trois fois jusqu'à ce que toutes les références internes soient à jour. Il est normal à ce stade que la figure 1 du document soit vide.

- Ajouter dans le préambule du fichier maître la commande

```
\includeonly{emacs}
```

puis compiler le document.

Observer que malgré l'absence du chapitre 1, la numérotation et les références demeurent à jour, notamment la table des matières.

- Remplacer la commande ajoutée en b) dans le préambule du fichier maître par la commande

```
\includeonly{presentation}
```

Vers la fin du fichier `presentation.tex`, activer la commande

```
\includegraphics[width=\textwidth]{console-screenshot}
```

en supprimant le symbole % au début de la ligne. Compiler de nouveau le document deux fois.

Les modifications ont eu pour effet d'ajouter une page au chapitre 1. Observer que selon la table des matières, le chapitre 2 débute toujours à la page 3 alors que celle-ci est maintenant occupée par la figure 1.

- d) Afin de corriger la table des matières, désactiver la commande `\includeonly` dans le préambule du fichier maître, puis compiler de nouveau le document quelques fois.

- 2.2 Déplacer dans un fichier `preamble.tex` toutes les lignes du préambule du fichier `exercice_include.tex` utilisé à l'exercice précédent, à l'exception de celles relatives à la page titre (titre, auteur, date). Insérer le préambule dans `exercice_include.tex` avec la commande `\input`.