



INTRODUCTION AU TRAITEMENT DE TEXTE L^AT_EX

OLIVIER BERNARD

Faculté des Sciences
Département de physique

6 janvier 2020



UNIVERSITÉ DE
SHERBROOKE

Table des matières

1	Mise en contexte	1
1.1	Qu'est-ce que L ^A T _E X	1
1.2	Comment cela fonctionne-t-il	1
2	Création d'un document et commandes de bases	1
2.1	Lignes de codes et commandes indispensables pour un fichier <i>.tex</i>	1
2.2	Fichier style, les fichiers <i>.sty</i>	2
2.3	Appelle de commandes, de fonctions et de symboles	3
3	Mise en forme du texte	3
3.1	Commandes de mise en forme	3
3.2	Équations & symboles	4
3.3	Références	6
4	Mise en page des images	6
4.1	Une seule image	6
4.2	Plusieurs images	7
5	Création de tableau	9
5.1	Tableau	9
6	Compléments	10
6.1	Section	10
6.2	En-tête	10
6.3	Liste à puce, numéros, etc.	12
6.4	Création de commandes	14
7	Annexes	I
7.1	Lettres grecs	I
7.2	Symboles mathématique	II
7.3	Code du fichier <i>.sty</i> de ce document	III

1 Mise en contexte

1.1 Qu'est-ce que L^AT_EX

L^AT_EX est un programme de traitement de texte (comme Word) qui ne fonctionne toutefois pas avec une interface graphique, mais avec des lignes de codes qui sont entrées à la main. Le texte est tapé dans une interface de programmation, puis son édition est faite en y entrant des lignes de codes.

Puisque L^AT_EX est un programme qui est plus près du code que de l'interface graphique, il est plus facile de contrôler le résultat final, mais son utilisation peut sembler moins intuitive.

1.2 Comment cela fonctionne-t-il

Pour débiter, il vous faut travailler soit localement sur votre ordinateur avec un compilateur (par exemple pour Linux il y a *TeXMaker*, pour Windows il y a *MikTeX* et pour Mac il y a *MacTeX*)¹ (vous pouvez toujours utiliser un éditeur mais il est préférable d'utiliser un compilateur). Soit en ligne avec OverLeaf (anciennement ShareL^AT_EX) .

Si vous travaillez sur votre ordinateur, vous devez télécharger les modules (angl. *package*) que vous désirez utiliser dans votre code avant de pouvoir les appeler dans un code².

Si vous travaillez en ligne, OverLeaf (ShareL^AT_EX) contient déjà tous les modules ainsi que toutes les bibliothèques de fonctions (vous devez tout de même les appeler dans votre code).

Une fois votre plateforme de travail déterminée, vous voudrez débiter un nouveau document (qui sera un fichier de format *.tex* par défaut, et ce peu importe la plateforme utilisée). Si vous travaillez sur OverLeaf (ShareL^AT_EX) , tous les nouveaux documents ont des commandes par défaut déjà écrites. Si vous travaillez sur votre ordinateur, il est possible que vous deviez entrer quelques lignes de code pour pouvoir débiter votre document (et qu'il puisse être compilé) (section 2.1 de ce document).

Une fois cela fait, vous pouvez commencer à taper votre document et à vous amuser à faire la mise en forme du texte et autres de documents (sections 2, 3, 4 & 5 de ce document)!

Si vous voulez utiliser les lignes de codes présentes dans ce document, il vous faut les écrire dans votre code et non pas les copier-coller.

Merci et bonne lecture.

2 Création d'un document et commandes de bases

2.1 Lignes de codes et commandes indispensables pour un fichier *.tex*

Au début de chaque document L^AT_EX, il vous faut tout d'abord spécifier la classe de votre document avec la commande `\documentclass`. Ensuite, il est possible de spécifier la taille de la police ainsi que d'autres informations (ceci est facultatif) entre crochets `[...]`. De plus, vous devrez spécifier la classe du document entre accolades `{...}` (par défaut, OverLeaf crée un document de classe *article* avec une taille de police prédéfinie). Par exemple :

1. Ce ne sont que quelques compilateurs parmi tant d'autres de disponibles en ligne (gratuitement)
2. Certains compilateurs téléchargent le module s'il n'est pas présent.

```
\documentclass[twoside,letterpaper,12pt]{article}
```

Ensuite, il est important d'utiliser les modules *utf8*, *fontenc* et *babel* qui permettent l'utilisation d'accent (et la reconnaissance de la langue française). Sans ces modules, les compilateurs L^AT_EX ne reconnaissent pas nécessairement les accents et il est donc possible que votre code ne compile pas. Ceci va également vous générer des messages d'erreurs. Voici les lignes pour appeler ces modules :

```
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}
```

D'ailleurs, vous voudrez utiliser le module *geometry* pour contrôler la taille des marges de vos pages :

```
\usepackage[top=2cm, bottom=2cm, left=2cm, right=2cm]{geometry}
```

Finalement, pour que le code L^AT_EX puisse compiler, il vous faut débiter et terminer le code. Pour ce faire, la première ligne de votre code avant le texte et contenu de votre document doit être³ :

```
\begin{document}
```

Ensuite, la dernière ligne de votre code devra être⁴ :

```
\end{document}
```

2.2 Fichier style, les fichiers *.sty*

Lors de la rédaction de documents L^AT_EX, vous serez appelés à utiliser quelques commandes qui proviennent de différentes bibliothèques de fonctions (appelés des *modules*). Ces *modules* contiennent les codes de commandes et de fonctions que L^AT_EX (et vous) utilise. Pour appeler des modules, vous devez utiliser la commande `\usepackage{}`. Vous devez entrer le nom du module entre les accolades (`{}`) (voir la section 2.1 pour avoir un exemple d'appel de module). Vous pouvez appeler une quantité de modules aussi grande que le nombre de modules disponible (en ligne ou sur votre ordinateur). Une autre façon de travailler avec les bibliothèques est d'utiliser des fichiers *.sty*. Ce genre de fichier est appelé un fichier *style* et est un code, comme votre *main.tex*, mais avec une fonction différente et avec l'extension *.sty*. Un des avantages des fichiers *.sty* est le fait que vous pouvez construire vous-même une banque de module que vous utilisez fréquemment (appeler tous les modules). Lorsque vous appelez votre fichier *.sty*, vous appelez également tous les modules qui y sont appelés (donc vous n'avez plus à appeler tous vos modules au début de chacun de vos code). Pour créer un tel fichier, vous devez tout d'abord créer un nouveau fichier (fichier *.sty*) dans lequel vous faite appelle a tous les modules que vous désirez utiliser dans vos codes L^AT_EX (voir la section 2.1 pour avoir un exemple d'appel de module). Une fois cela fait, vous devez enregistrer sous le fichier avec l'extension *.sty*. Somme toute, c'est une bibliothèque de modules et de fonctions faite par vous-même (ou quelqu'un d'assez gentil pour

3. Il peut y a voir du code avant cette ligne, tel les appels de modules, bibliothèques, etc.

4. Sans exception, cette ligne doit être la dernière du document

partager son dur labeur). Toutefois, tout comme les *modules* réguliers, vous devez appeler votre bibliothèque *.sty*. Pour ce faire, vous devez, encore une fois, utiliser la fonction `\usepackage{}` pour l'appeler (notez que vous devez entrer dans les crochets `{}` seulement le nom de votre fichier *.sty* sans son extension).

2.3 Appelle de commandes, de fonctions et de symboles

Pour faire appel à une commande ou un symbole quelconque, il faut toujours utiliser le symbole `\` (le *backslash*) pour que L^AT_EX puisse le reconnaître comme tel (vous devez écrire le nom de la commande, fonction ou symbole suite au `\`). Lorsque vous utiliser le `\`, vous verrez le texte qui lui est collé changer de couleur dépendamment de la nature de ce que vous écrivez (commande, fonction ou symbole).

3 Mise en forme du texte

3.1 Commandes de mise en forme

Dans L^AT_EX, pour faire de la mise en forme de texte, il suffit de faire appel à la/les commande(s) appropriée(s) pour faire la mise en forme que vous désirez. Par exemple, la commande `\textbf{}` est utilisée pour mettre le texte en **gras** (`\textit{}` pour *italique* et `\underline{}` pour souligner).

Lorsque vous écrivez du texte et que vous avez terminé votre phrase, mais pas votre ligne, et que vous voulez sauter une ligne, vous voudrez faire un retour de chariot (un *enter*). Pour ce faire, vous n'avez qu'à utiliser la commande `\\` à la fin de votre ligne. Par exemple :

J'écris un court paragraphe \\ En deux segments.

Donne :

J'écris un court paragraphe
En deux segments.

Ensuite, il peut arriver que vous ayez besoin d'espacer certains mots ou objets dans votre document. Une façon d'imposer un espace entre deux objets est d'utiliser le `\quad` ou `\qquad`. Le premier étant un espace un peu plus long qu'un espace régulier, et le second un grand espace. Une autre façon est d'utiliser simplement «`\,`». Par exemple :

Un `\quad` espace plus `\qquad` grand.

Donne :

Un espace plus grand.

Il est aussi possible de décider de modifier la taille du texte pour un mot, ou une phrase spécifique. Pour ce faire, une fonction entre accolade qui est le `{\size}` peut être utilisée. Pour utiliser cette commande, vous devez mettre le segment de texte que vous désirez modifier entre accolade (`{}`) à la suite de la commande `{\size}`. Vous pouvez utiliser 3 tailles (*size*) : gros, par défaut (celle standard de votre document) et petit. Par exemple, le code :

```
{\large texte} \\
{\normalsize texte} \\
{\tiny texte} \\
```

Donne ceci :

```
texte
texte
texte
```

Finalement (dans cette introduction), il est possible de surligner du texte ainsi que de choisir la couleur du surlignage. Pour faire ceci, vous aurez besoin du package *color*. Pour faciliter l'utilisation de cette fonction, il est suggéré⁵ :

```
\usepackage{soul}
\usepackage{color}
\newcommand{\highlight}[1]{\colorbox{lightgray}{#1}}
```

Pour surligner du texte, vous n'avez qu'à taper la commande `\highlight{}` (vous n'avez qu'à taper le texte que vous désirez surligner entre les accolades). Pour modifier les paramètres de la couleur du surlignage, vous devez taper le nom de la couleur désirez (en anglais) entre les accolades après le segment `\colorbox` dans la ligne de code fourni ainsi que le numéros après le dièse (le #) entre accolade.

3.2 Équations & symboles

Dans plusieurs domaine de la science et du travail, les rédacteurs sont souvent appelés à écrire des équations mathématiques qui peuvent être simples ou bien complexes. Pour ce faire, il faut utiliser quelques modules (les suivants) :

```
\usepackage{amsmath,amssymb,bm,upgreek}
```

Lors de l'écriture de fonctions mathématiques, généralement, il est possible de procéder de deux façons : la première étant de les introduire dans le texte, la seconde étant de les mettre en évidence à l'aide de fonctions appropriées.

Lorsqu'un symbole⁶ doit être introduit dans une formule, il est recommandé de procéder de la même manière que spécifié dans la section 2.3 (par exemple `\omega` donne ω). Notez que si vous mettez la première lettre de la variable en majuscule, vous obtiendrez le même symbole, mais en majuscule⁷ (merci de ne pas m'en vouloir pour la redondance).

Lors de la rédaction d'équations mathématiques dans le texte, il faut entourer l'équation (ou bien le segment de texte ou de code désiré) par une paire de `$`. L'utilisation de ces symboles créer un environnement dit *mathématique* dans le code qui impose la typographie mathématique au segment que la paire de `$` entoure (cette typographie élimine les espaces ainsi que les accents). De

5. Notez que ceci est une fonction qui a été créé dans L^AT_EX et qui doit être écrite soit dans votre fichier *.sty* ou avant votre *begin{document}*

6. Un tableau des lettres Grec ainsi qu'un tableau de symboles mathématiques qui sont souvent utilisé sont présent à la section 7

7. Si le symbole à une majuscule bien entendu

plus, sans l'utilisation des symboles \$, l'ajout d'indices et d'exposants est possible mais génère des erreurs majeures.

Ensuite, il faut souvent exprimer des exposants ainsi que des indices pour les variables dans différentes équations. Pour cela, il faut utiliser le caractère `_` pour mettre un indice et utiliser le caractère `^` pour mettre un exposant à une variable⁸. Finalement, le tout doit être entouré par une paire de \$⁹. Par exemple :

```
$R_{e}^{P}$ \
```

Les indices `$_{Ici}$` et les exposants `$_{La}$`, génial `$_{Bas}^{Haut}$`

Donne :

$$R_e^P$$

Les indices $_{Ici}$ et les exposants La , génial $_{Bas}^{Haut}$

Pour écrire une fraction dans une équation, la fonction `\frac{}{}` peut être utilisé. La première paire d'accolade est le numérateur de la fonction et la seconde paire est le dénominateur de la fraction. Par exemple :

```
$ c = \frac{a}{365-b}$
```

Donne :

$$c = \frac{a}{365-b}$$

Lorsque les équations sont en évidence, l'utilisation de la fonction `\begin{align}` est suggéré. Cette fonction permet de numéroté, d'aligner (vous devez mettre le symbole `&` pour spécifier l'endroit où vous voulez que l'alignement soit fait) ainsi que d'écrire en typographie mathématique (cette fonction crée également un environnement mathématique). Voici deux exemples de formules mise en évidence à l'aide de la fonction `\begin{align}` :

Par exemple¹⁰ :

```
\begin{align}
T_{R} &= \frac{R}{\sqrt{\left(R+R_{L}\right)^2}} \\
&\label{frm} \\
\end{align}
```

Donne :

$$T_R = \frac{R}{\sqrt{(R + R_L)^2}} \quad (1)$$

Voici une autre formule¹¹ :

8. Lorsque vous mettez des indices et des exposants, mettez ce que vous désirez mettre en indice ou exposant entre accolades

9. Sans ces symboles vous ne travaillez pas dans un environnement dit *mathématique*

10. Il est aussi possible d'utiliser la fonction *gather*

11. Notez l'utilisation de `\nonumber` pour éliminer le numéros de l'équation. Si vous voulez écrire des équations sans qu'elles ne soient numérotées, utilisez * suite à *align* ou *gather*


```

\begin{align}
\nonumber
T_{c} = \frac{1}{\omega C \sqrt{\left(R + R_{L}\right)^2}}
\label{frm_2}
\end{align}

```

Donne :

$$T_c = \frac{1}{\omega C \sqrt{(R + R_L)^2}}$$

3.3 Références

Dans les textes, il est possible de faire référence aux formules, images, tableaux, etc qui ont été identifiées à l'aide d'un *label*. Pour ce faire, il vous faut le module *hyperref* :

```
\usepackage[hidelinks]{hyperref}
```

Ensuite, il faut écrire `\label{}` sous l'objet (équations, image, etc) qui doit être identifié, puis lui attribuer un nom de référence dans l'accolade. Ensuite, pour faire référence au label (ou l'objet identifié), la fonction `\ref{}` doit être employée. Par exemple, si vous voulez faire référence à la formule 1, il est possible d'utiliser la fonction `\ref{}` ou `\eqref{}` pour avoir le mot *équation* devant le numéros de référence. Une fois le `\ref{}` (ou `\eqref{}`) utilisé, le *label* de l'objet désiré doit être écrit dans l'accolade. Par exemple :

```

\ref{frm}
\eqref{frm}

```

Donne :

1
équation (1)

Ce qui correspond au numéros de l'équation de T_R . L'avantage de l'utilisation des *label*, est la mise à jour automatique des références (son numéros). Le même principe s'applique aussi aux images et graphiques, qui seront traités un peu plus loin dans ce document ¹².

4 Mise en page des images

4.1 Une seule image

Pour faire la mise en page d'images, les modules *caption*, *float* et *graphicx* sont requis :

```

\usepackage{caption}
\usepackage{graphicx}
\usepackage{float}

```

12. Pour faire référence à une image (même si elle fait partie d'un ensemble), nous n'avons qu'à appliquer le même principe que pour les équations. Par exemple, `\ref{im4}` fait référence à l'image 2c de la sous-section 4.2, ce qui correspond à l'image des points des initiés.

Ensuite, il faut les appeler puis donner quelques informations de base¹³. Tout d'abord, pour emmener une image dans le document, il faut employer la fonction `\begin{figure}`. Une fois la figure commencée, elle peut être centrée avec la fonction `\centering`, puis il est nécessaire d'appeler l'image désirée à l'aide de la fonction `\includegraphics[width=qqc]{nom}`. Dans cette fonction, il faut spécifier la taille de l'image entre crochet ainsi que le nom de l'image (avec son extension) entre accolades. Ensuite, il est possible de nommer la figure avec `\caption{}` et enfin fermer la figure avec un `\end{figure}`.

Par exemple :

```
\begin{figure}[ht]
  \centering
  \includegraphics[width=100mm]{MBC.jpg}
  \caption{Mon beau chien \textit{Gargouille}}
  \label{im}
\end{figure}
```

Donne :



FIGURE 1 – Mon chien *Gargouille*

4.2 Plusieurs images

Ensuite, pour créer un groupe d'image, il vous faut tout d'abord ajouter le module *subcaption* aux modules précédents (voir le début de la section 4) :

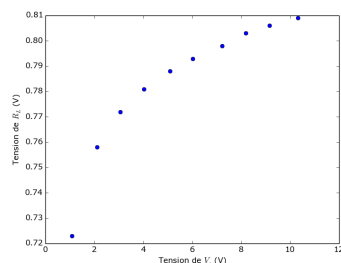
```
\usepackage{caption}
\usepackage{graphicx}
\usepackage{subcaption}
```

13. Si vous travaillez sur OverLeaf, vous devrez importer vos images en ligne dans votre projet pour pouvoir les utiliser. Sur votre ordinateur, les images doivent être dans le même document que votre code.

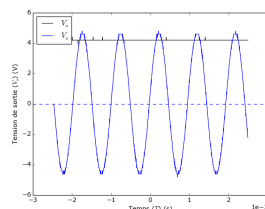
Ensuite, il faut employer les mêmes principes que pour une image seule. Il faut débiter avec un `\begin{figure}`. Une fois la figure commencée, il est possible de la centrer avec la fonction `\centering`, puis une sous-figure doit être créée (`\begin{subfigure}`, qui est semblable à la figure) puis dans cette sous-section, `\includegraphics` est employé de nouveau¹⁴. Dans cette fonction, la taille de l'image peut être contrôlée en la spécifiant entre les crochets. Aussi il faut donner le nom de l'image (avec son extension) entre accolade. En ajout, il est possible de lui donner un sous-nom avec la fonction `\caption{}` et la sous-figure est fermée avec un `\end{subfigure}`. Si vous voulez donner un *label* à une sous-image, vous n'avez qu'à écrire le code `\label{}` sous le *caption* de l'image et procéder comme spécifié dans la section 3.3. Le processus doit être répété pour chaque image que vous voulez (dans l'exemple qui suis ce sera pour trois images). Puis, il est possible de donner un noms à l'ensemble d'image qui vient d'être introduite à l'aide d'un dernier `\caption` qui n'est pas dans une sous-image. Une fois cela fait, l'image doit être fermée avec un dernier `\end{figure}`. Par exemple :

```
\begin{figure}[ht]
  \centering
  \begin{subfigure}[b]{50mm}
    \includegraphics[width=\textwidth]{beaugraphique.png}
    \caption{Graph 1}
    \label{im2}
  \end{subfigure}
  \begin{subfigure}[b]{40mm}
    \includegraphics[width=\textwidth]{beaugraph2.png}
    \caption{Graph 2}
    \label{im3}
  \end{subfigure}
  \begin{subfigure}[b]{30mm}
    \includegraphics[width=\textwidth]{points.jpg}
    \caption{Points des initiés}
    \label{im4}
  \end{subfigure}
  \caption{Exemple de disposition d'images}
\end{figure}
```

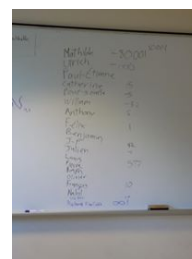
Donne :



(a) Graph 1



(b) Graph 2



(c) Points des initiés

FIGURE 2 – Exemple de disposition d'images

14. La commande `include graphics` a un `caption` de `graphic` et `include figures` un `caption` de `figure`.

5 Création de tableau

5.1 Tableau

Dans \LaTeX , faire des tableaux simple peut s'avérer être à la fois une tâche simple et ardue. Tout d'abord, cette tâche peut être fastidieuse car il faut entrer les données manuellement et les mises à jour ne se font pas automatiquement (donc il est préférable de soit se faire un tableau vide puis le remplir ou en faire un une fois que toutes vos données sont acquise). Elle peut être conviviale car la forme exacte du tableau est connue.

Premièrement, pour créer un tableau, il vous faut le module *array* :

```
\usepackage{array}
```

Et utiliser la ligne de code suivante (pour que le titre sois *Tableau* et non *Table*) :

```
\addto\captionsfrench{\def\tablename{\sc Tableau}}
```

Ensuite, il vous faut utiliser la fonction `\begin{table}[ht]`. Une fois cela fait, il est possible de le centrer à l'aide d'une commande `\centering`, ce qui permet de débiter le le *tabular*. Avec la fonction *tabular*, il faut spécifier, entre accolades, le nombre de colonnes (avec des **c**) ainsi qu'où se situent les lignes verticales du tableau (avec des "|") (voir l'exemple pour plus de détails). Puis, pour spécifier les lignes horizontales, la commande `\hline`¹⁵ est utilisée à l'endroit où une ligne horizontale doit se situer. Une fois la tête du tableau fait, il faut employer des `\\` pour faire un retour de chariot (descendre d'une ligne). N'ensuite, l'emploi des éperluettes (`&`) est nécessaire pour spécifier la séparation des colonnes¹⁶. Enfin, il faut terminer le tableau avec un `\end{tabular}` et un `\end{table}`. Par exemple :

```
\begin{table}[ht]
  \centering
  \begin{tabular}{|c c |c c c|}
    \hline
    Composantes & & & Valeurs & & \\
    \hline \hline
    $R_3$ & ( $\Omega$ ) & 9960 &  $\mu$  & 2 & \\
    \hline
    $R_4$ & ( $\Omega$ ) & 198,6 &  $\mu$  & 0,1 & \\
    \hline
    $R_5$ & ( $\Omega$ ) & 99600 &  $\mu$  & 10 & \\
    \hline
    $C_1$ & (F) & 199,07 &  $\mu$  & 0,01 & \\
    \hline
  \end{tabular}
  \caption{Valeurs des Composantes de la partie 2 }
\end{table}
```

Donne :

15. pour faire une double ligne horizontale, vous devez mettre deux fois la commande `\hline` de suite.

16. Le nombre d'éperluette dans chaque ligne doit équivaloir au nombre de colonnes (les **c**) du tableau (moins un) spécifié dans la commande *tabular*

Composantes		Valeurs		
R_3	(Ω)	9960	\pm	2
R_4	(Ω)	198,6	\pm	0,1
R_5	(Ω)	99600	\pm	10
C_1	(F)	$199,07 \cdot 10^{-9}$	\pm	0,01

TABLEAU 1 – Valeurs des Composantes de la partie 2

6 Compléments

6.1 Section

Pour créer une section, simplement utiliser la fonction `\Section{}`

Pour un sous-section, utiliser la fonction `\Subsection{}`

Pour faire un autre niveau de section, utiliser la fonction `\Subsubsection{}`

Spécifier le nom de votre section, sous-section ou sous-sous-section entre les accolades.

Cela va faire le compte des sections de façon automatique et en faire les mises à jours.

6.2 En-tête

Pour faire des en-têtes comme dans ce document, il vous faudra quelques petits ajouts. Tout d'abord, il vous faut le module *fancyhdr*. Ensuite, il faut spécifier le style de page comme étant *fancy*. À la suite de cela, il vous faut spécifier le texte qui sera écrit aux différents endroit en appelant un fancyhead (`\fancyhead`). Pour cette commande, vous devez spécifier l'emplacement dans l'en-tête (*L* à gauche, *R* à droite et *C* au centre)¹⁷, entre crochets [...] et vous devrez écrire entre accolades le texte désirée {...}. Voici en exemple le code utilisé pour l'en-tête de ce document :

```
\usepackage{fancyhdr}

\pagestyle{fancy}

\renewcommand{\headrulewidth}{0.5pt}
\fancyhead[C]{}
\fancyhead[LO]{\textbf{Section \, \thesection}}
\fancyhead[LE]{\rightmark}
\fancyhead[RO]{\rightmark}
\fancyhead[RE]{\textbf{Section \, \thesection}}
```

Pour faire un pied de page, le code est identique à l'exception que vous devez écrire *fancyfoot* au lieu de *fancyhead*. Notez que ces lignes de code doivent être écrite avant le `\begin{document}`¹⁸. Voici en exemple le code qui était utilisé pour le pied de page de ce document :

17. *RO* a droite des pages impaires, *RE* a droite des pages paires, *LO* a gauche des pages impaires et *LE* a gauche des pages paires

18. Il est déconseillé d'écrire cette commande dans un fichier *.sty*. Il est plus facile de travailler avec cette fonction dans votre fichier *main.tex*

```
\renewcommand{\footrulewidth}{0.5pt}
\fancyfoot[C]{}
\fancyfoot[LE]{}
\fancyfoot[LO]{\textbf{page} \thepage}
\fancyfoot[RE]{\textbf{page} \thepage}
\fancyfoot[RO]{Introduction \LaTeX}
```

6.3 Liste à puce, numéros, etc.

Dans les document \LaTeX , il est parfois faire requis de faire des listes (à puce, numéroté, etc). Pour ce faire, il faut tout d'abord appeler le package *enumitem* :

```
\usepackage{enumitem}
```

Une fois ce package appelé, nil est possible de créer des listes composé de différents symboles, chiffres, alphabets, etc. Pour créer une liste a puces standard, il faut tout d'abord créer un environnement *liste* à l'aide de `\begin{itemize}[label=... , font= size]`. Entre les accolades, il sera spécifié le type de liste désiré (*itemize* ou *enumerate*). Ensuite, entre les crochets, il est possible de définir ce qui servira de puce. Finalement, la taille des points peut être définit. Si les crochets sont vides, \LaTeX utilise un symbole et une taille par défaut. Par exemple :

```
\begin{itemize}
  \item Premier
  \item Second
\end{itemize}
```

Donne :

- Premier
- Second

Lorsque un *label* et un *font* sont définit entre les crochets, cela donne une nouvelle forme à la liste. Par exemple :

```
\begin{itemize}[label=\textbullet, font=\scriptsize]
  \item Premier
  \item Second
\end{itemize}
```

Donne ceci :

- Premier
- Second

De plus, dans les options de *enumitem*, un label vide (`{}`) peut être donné, ou simplement un espace. Par exemple :

```
\begin{itemize}[label={}]
  \item Premier
  \item Second
\end{itemize}
```

Donne :

- Premier
- Second

Finalement (pour les `itemize`), un symbole ou autre choses peut être défini comme *label* dans les listes. Par exemple :

```
\begin{itemize}[label=$\lambda$, font=\scriptsize]
  \item Premier
  \item Second
\end{itemize}
```

Donne :

- λ Premier
- λ Second

La seconde option pour les listes sont les listes *enumerate*. Lorsque cette option est utilisée, il est possible de définir différent *label* (*Numeros*, *ChiffresRomain*, *Alphabet*, etc). En plus, il y a la possibilité d'imbriquer des listes dans des listes (chaque imbrication va créer un indentation entre la liste imbriqué et la liste qui lui est supérieur). En plus d'énumérer et de mettre à jour les liste automatiquement, les sous-sections s'incrémente aussi avec la liste dans laquelle elles sont introduite (à l'exception des listes avec les alphabets). S'il n'y a pas de *label* spécifié, la liste par défaut utilisé par L^AT_EX sera la liste à numéros standard. Par exemple :

```
\begin{itemize}[label=\textbullet, font= \scriptsize]
  \item Premiere
  \begin{enumerate}
    \item Sous-premier
    \begin{enumerate}[label=\Alph*]]
      \item Ss-sous premier
      \begin{enumerate}[label=\alph*]]
        \item Ss-sous premier
      \end{enumerate}
    \item Ss-sous second
    \begin{enumerate}
      \item Ss-ss-sous premier
    \end{enumerate}
  \end{enumerate}
  \item Sous second
  \begin{enumerate}
    \item Ss-sous Premier
  \end{enumerate}
\end{itemize}
```

Donne :

- Premier
 - 1. Sous-premier
 - A) Ss-sous premier
 - a) Ss-sous premier

- B) Ss-sous second
 - i. Ss-ss-sous premier
- 2. Sous second
 - 2.1. Ss-sous Premier

6.4 Création de commandes

Dans cette sous-section, la création de commandes sera traitée et expliqué. Tout d'abord, comme mentionné plus haut (voir la sous-section 3.1) la commande qui permet de surligner (le *highlight*) du texte à été crée par l'auteur de ce document :

```
\newcommand{\hilight}[1]{\colorbox{lightgray}{#1}}
```

Cette commande est fort simple de création ainsi que d'utilisation. Pour créer une commande, il faut bien entendu commencer le tout par

`\newcommand{\nom}[# entrer]{\quoi{#x commande #x2 commande etc.}}`. Cette simple ligne de commande comporte quatre sections. La première section (`{\nom}`) est utilisée pour donner le nom de la commande créée¹⁹(par exemple `{\hilight}`) utilisé dans un fichier *.tex* (par exemple `\hilight{mot surligné}`). La seconde section (le `[# entrer]`) est employée pour indiquer le nombre d'éléments/d'entrées que la commande employe/utilise. Dans le cas de la commande démontrée plus haut, une seule entrée est requise et c'est le segment qui sera surligné. La troisième section elle (le `{\quoi{...}}`) est utilisé pour indiquer comment la commande sera appelée et/ou employée dans d'autres commandes s'il y a présence de `\def` avant (voir prochain exemple). Sinon, ce n'est simplement que l'appel d'une commande existante (par exemples `{\colorbox{lightgray}{...}}`). Enfin, la quatrième section (le `{...#x commande #x2 commande etc.}}`) est employée pour indiqué où et comment chaque élément requis par la commande sera utilisée. Dans le cas de la commande mentionnée plus haut, une seule entrée est requise et elle est employée immédiatement. Dans des cas plus complexe (dans le prochain exemple), ces entrées peuvent changées, déplacées, êtres altérées, etc.

Passons à un exemple plus complexe. Voici un extrait du code de la bibliothèque de commandes de ce document qui sert à créer des entêtes de devoir :

```
%Faire un en-tête de devoir
\newcommand*\redacteur}[3]{\def\Nomredacteur{#1 \hfill #3 \\\ #2}}
\newcommand*\coteCours}[1]{\def\CoteDuCours{#1}}
\newcommand*\nodev}[1]{\def\NoDev{#1}}
\newcommand\entete{
  \thispagestyle{firststyle}
  \flushleft
  \begin{minipage}{17cm}
  {\Large \Nomredacteur \par}
  \vspace{2mm}
  {\large \CoteDuCours}
  \end{minipage}
```

19. S'il y a une commande déjà existante avec le nom que vous désirez utiliser, il est suggéré de changer le nom de votre commande ou de redéfinir la commande (ce qui n'est pas traité ici).

```
\vspace{0.5cm}
\headline{\textbf{{\Large DEVOIR \NoDev}}}
\vspace{1cm}
}
%Commande pour faire une en-tête:
%\redacteur{NOM1}{NOM2}{\today}
%\coteCours{CODE DU COURS}
%\titreDuCours{NOM DU COURS}
%\nodev{Numéros du devoir}
%\entete
```

Vous noterez que la nouvelle commande démontrée est accompagnée des commandes qui doivent être écrites (après le `\begin{document}`) et utilisées par l'auteur d'un document pour appeler et compléter la dite commande. Les trois premières `\newcommand` respectent le format et les explications mentionnées lors de l'explication de la première création de commande (remarquez l'emploi des `\def`). Toutefois, je vous invite à noter le format particulier de la première commande : il y a trois entrées (`#1`, `#2` et `#3`) qui sont disposés de la façon suivante : `{#1 \hfill #3 \ \ #2}` pour permettre l'écriture des noms complet de deux auteurs sur une même colonne et la date à droite de la page. Pour appeler cette commande et la compléter, il faut faire l'appel suivant : `\redacteur{NOM1}{NOM2}{\today}`.

Ensuite, il y a la quatrième commande qui ne respecte pas la structure présentée plus tôt. En effet, cette commande est plus complexe car elle appelle, positionne et altère les commandes créées avant elle. Tout d'abord, dès l'appel de la création de commande (`\newcommand`), le nom de la fonction est énoncé (`\entete`) puis la commande est ouverte avec une première accolade. Entre l'accolade d'ouverture et de fermeture, toutes les lignes d'appels et de commandes sont écrites pour cette nouvelle fonction. Cette commande appelle d'abord un style de page (`\thispagestyle{firststyle}`), une justification du texte (`\flushleft`), puis crée un environnement de travail (`\begin{minipage}{17cm}`). Suite à cela, une première commande est appelée (`{\Large \Nomredacteur \par}`). Cet appel indique où dans l'environnement cette commande sera localisée²⁰. Ensuite, une commande de mise en page (`\vspace{2mm}`) a été ajoutée pour séparer les deux commandes appelées. Ensuite une seconde commande est appelée (`{\large \CoteDuCours}`) et l'environnement de travail est fermé (`\end{minipage}`). Enfin, une autre commande de mise en page (`\vspace{0.5cm}`) est écrite, suivie de la dernière commande appelée (`\headline{\textbf{{\Large DEVOIR \NoDev}}}`) et n'est pas limitée dans un espace de travail défini mais plutôt par la feuille. Enfin, la fonction appelle une dernière commande de mise en page (`\vspace{1cm}`) pour séparer le texte qui suit la commande et la commande en soit avant d'être clôturée par une dernière accolade.

Enfin, pour appeler la commande dans un document et l'utiliser correctement, il faut que les lignes de codes (ainsi que les modules requis pour la faire fonctionner) soient écrites avant le `\begin{document}` ou dans un fichier `.sty`. En plus, il faut lui donner les entrées nécessaires à l'aide des lignes d'entrées que vous (ou l'auteur de la commande) avez créées. Par exemple, `\redacteur{NOM1}{NOM2}{\today}` donne les entrées employées par `\Nomredacteur` (dans les accolades). Ou encore, `\coteCours{CODE DU COURS}` donne l'entrée employée par `\CoteDuCours`.

20. Remarquez qu'il n'y a pas de besoin quant à ce qui attire à spécifier des détails supplémentaires par rapport aux détails qui ont déjà été spécifiés dans la commande appelée

7 Annexes

7.1 Lettres grecs

Nom	Symbole ou commande	Résultat
alpha	<code>\alpha</code>	α
beta	<code>\beta</code>	β
gamma	<code>\gamma</code>	γ
delta	<code>\delta</code>	δ
epsilon	<code>\epsilon</code>	ϵ
zeta	<code>\zeta</code>	ζ
eta	<code>\eta</code>	η
theta	<code>\theta</code>	θ
kappa	<code>\kappa</code>	κ
lambda	<code>\lambda</code>	λ
mu	<code>\mu</code>	μ
nu	<code>\nu</code>	ν
xi	<code>\xi</code>	ξ
pi	<code>\pi</code>	π
rho	<code>\rho</code>	ρ
sigma	<code>\sigma</code>	σ
tau	<code>\tau</code>	τ
upsilon	<code>\upsilon</code>	υ
phi	<code>\phi</code>	ϕ
chi	<code>\chi</code>	χ
psi	<code>\psi</code>	ψ
omega	<code>\omega</code>	ω
Gamma	<code>\Gamma</code>	Γ
Delta	<code>\Delta</code>	Δ
Theta	<code>\Theta</code>	Θ
Lambda	<code>\Lambda</code>	Λ
Xi	<code>\Xi</code>	Ξ
Pi	<code>\Pi</code>	Π
Sigma	<code>\Sigma</code>	Σ
Upsilon	<code>\Upsilon</code>	Υ
Phi	<code>\Phi</code>	Φ
Psi	<code>\Psi</code>	Ψ
Omega	<code>\Omega</code>	Ω
varepsilon	<code>\varepsilon</code>	ε
vertheta	<code>\vartheta</code>	ϑ
varpi	<code>\varpi</code>	ϖ
varrho	<code>\varrho</code>	ϱ
varsigma	<code>\varsigma</code>	ς
varphi	<code>\varphi</code>	φ

7.2 Symboles mathématique

Nom	Symbole ou commande	Résultat
Infinie	<code>\infty</code>	∞
Dérivé partielle	<code>\partial</code>	∂
Intégrale	<code>\int</code>	\int
Intégrale curviligne	<code>\oint</code>	\oint
plus ou moins	<code>\pm</code>	\pm
Angle	<code>\angle</code>	\angle
Somme	<code>\sum\limits^ {Dessus} _{Dessous}</code>	$\sum_{Dessous}^{Dessus}$
racine	<code>\sqrt[ordre]{dsRacine}</code>	$\sqrt[ordre]{dsRacine}$
vecteur	<code>\overrightarrow{ abc }</code>	\overrightarrow{abc}
Flèche vers la droite	<code>\rightarrow</code>	\rightarrow
Flèche vers la gauche	<code>\leftarrow</code>	\leftarrow
Flèche vers la gauche et droite	<code>\Leftrightarrow</code>	\Leftrightarrow
Inégale	<code>\neq</code>	\neq
Équivalent	<code>\equiv</code>	\equiv
Environ égale	<code>\simeq</code>	\simeq
Environ	<code>\sim</code>	\sim
Approximatif	<code>\approx</code>	\approx
Nabla	<code>\nabla</code>	∇
Convolution	<code>\ast</code>	$*$
Réel	<code>\Re</code>	\Re
Imaginaire	<code>\Im</code>	\Im
Produit scalaire	<code>\cdot</code>	\cdot
Produit vectorielle	<code>\times</code>	\times
C.Q.F.D.	<code>\Box</code>	\square
Lagrange ²¹	<code>\mathscr{L}</code>	\mathcal{L}
Hamiltonien	<code>\mathscr{H}</code>	\mathcal{H}

21. il vous faut le module *mathrsfs* pour pouvoir écrire ce symbole

7.3 Code du fichier *.sty* de ce document

Bien que le code soit fonctionnel, il n'est présent qu'à titre d'exemple. Certaines lignes ont été tronquées sur deux lignes pour pouvoir entrer sur le format de page de ce document.

Si vous voulez utiliser ce code, il vous ai suggéré de le retranscrire dans un fichier indépendant, ou de le télécharger sur le site <https://github.com/Olivier-L-Bernard> .

```
%PACKAGE -----
\usepackage[T1]{fontenc}      % un second package
\usepackage[top=2cm, bottom=2cm, left=2cm, right=2cm]{geometry} %pour les marges

\usepackage{lmodern}

\usepackage[super, square]{natbib}
\usepackage[utf8]{inputenc} % latin1      % Support de Latex pour les accents
\usepackage[french]{babel}              % Support du pdf pour les accents

\usepackage{fancyhdr} %pour les en_tetes et bas_de_pages

\usepackage{float}
\usepackage{newfloat}

\usepackage{ragged2e}

\usepackage{eso-pic}

%\AddThinSpaceBeforeFootnotes
%\FrenchFootnotes %notes

\usepackage{cancel}
\usepackage{physics}
%\usepackage[per-mode=symbol]{siunitx}
\usepackage[per-mode=symbol, separate-uncertainty = true]{siunitx}
%\SI{Nombre}{Unité}
% \per donne fraction \
% nombre donne x 10^{nombre}

\usepackage{enumitem} %pour les liste à puces

\usepackage{soul} %Permet de surligner du texte

\usepackage{mathrsfs} %Le symbole de Lagrange

%\usepackage{wasysym}
\usepackage{marvosym} %symbole de téléphone et autres

\usepackage{color}
```

```

%Permet de créer des liens dynamiques dans le document
\usepackage[hidelinks]{hyperref}

%Permet d'insérer des bouts en format paysage
\usepackage{pdflscape}

\usepackage{esint}

%Pour insérer du code pis que ce soit beau
\usepackage[most]{tcolorbox}

\usepackage{caption} %Permet de placer des libellés sous les tableaux, Figure, etc
\usepackage{subcaption} %Permet de placer des sous-libellés dans des sous-figures, etc

\usepackage{setspace} %pour les interlignes

\usepackage{array} %pour les tableaux
\usepackage{amsmath,amssymb,bm,upgreek}
\usepackage{tikz}
\usepackage{graphicx}

%Couleur personnalisées
\usepackage{xcolor} %permet de créer des couleurs personnalisés

\usepackage{titlesec} %Pour ligne entre section

\usepackage[font={small}, justification=centering]{caption} %Permet de placer des libellés

\def\headline#1{\hbox to \hsize{\hline{3cm}\quad\lower.3em\hbox{#1}\quad\hrulefill}} %Permet de créer des lignes de titre

%COMMANDE DE PAGE ET STYLE -----
\fancypagestyle{firststyle} %nouvelle commande pour modifier en-tête d'une page (spécifié
{
  \renewcommand{\headrulewidth}{0pt}
  \fancyhf{}
  \renewcommand{\headrulewidth}{0pt}
  \renewcommand{\footrulewidth}{0pt}
  \fancyfoot[C]{\thepage}
}

\fancypagestyle{firststyle2} %nouvelle commande pour modifier en-tête d'une page (spécifié
{
  \renewcommand{\headrulewidth}{0pt}
  \renewcommand{\footrulewidth}{0pt}
  \fancyhf{}
}

```

```
%ENVIRONNEMENT-----

\def\headline#1{\hbox to \hsize{\hrulefill\quad\lower.3em\hbox{#1}\quad\hrulefill}}

\DeclareFloatingEnvironment[name={Graphique}]{graph}
\DeclareFloatingEnvironment[name={Tableau}]{tableau}
\DeclareFloatingEnvironment[fileext=lop]{graphique}
\DeclareFloatingEnvironment[fileext=lop]{schema}
\addto\captionsfrench{\def\figurename{\sc Figure}} %Change le libellé de Figure à Figure
\addto\captionsfrench{\def\tablename{\sc Tableau}}
\addto\captionsfrench{\def\schemaname{\sc Schéma}}

%COMMANDES -----

\usepackage{etoolbox}
\appto\UrlBreaks{\do\~}

\newcommand\sidecomment[5][0.1]%
{\begin{tikzpicture}[remember picture,overlay]
  \draw[-stealth]
    ($({pic cs:#4}|-{pic cs:#2})+(#1,0)$)
    .. controls +(0.2,-0.05) and +(0.2,0.1) ..
    node[right,align=left]{#5}
    ($({pic cs:#4}|-{pic cs:#3})+(#1,0.1)$);
\end{tikzpicture}%
}

\renewcommand{\labelenumii}{\theenumii}
\renewcommand{\theenumii}{\theenumi.\arabic{enumii}.}

\renewcommand{\bibsection}{}

%Commandes permettant de référencer plus simplement les figures et équation
\newcommand{\figref}[1]{figure (\ref{#1})} %Figure
\renewcommand{\eqref}[1]{équation (\ref{#1})} %Équation

\newcommand{\highlight}[1]{\colorbox{lightgray}{#1}} %Autre méthode de surlignage

\DeclareMathOperator{\sinc}{sinc}

%\newcommand*\citen[1]{%
% \begingroup
% \romannumeral-\x % remove space at the beginning of \setcitestyle
% \setcitestyle{numbers}%
% \cite{#1} %
% \endgroup
```

```

%}

\newcommand\Ord{\mathcal{O}}
\newcommand\eqH{\stackrel{\mathclap{\normalfont{H}}}{=}}

\sisetup{detect-weight=true,detect-inline-weight=math}
\sisetup{inter-unit-product =\ensuremath{\cdot}}

% notation de la moyenne
\newcommand {\aqty}[1]{\langle #1 \rangle}

% matrice identité avec option de ne pas avoir 1 sur la diagonale
\newcommand{\idmat}[2][1]
{
  {
    {
      \newtoks\matrixtoks
      \global\matrixtoks = {}
      \newcount\rowcount
      \newcount\colcount
      \loop
      \colcount = 0
      \advance \rowcount by 1
      {
        \loop
        \advance \colcount by 1
        \edef\addtoks
        {
          \ifnum \colcount = 1 \else & \fi
          \ifnum \colcount = \rowcount #1 \else 0 \fi
        }
        \global\matrixtoks = \expandafter{\the\expandafter\matrixtoks\addtoks}
        \ifnum \colcount < #2
        \repeat
      }
      \ifnum \rowcount < #2
        \global\matrixtoks = \expandafter{\the\matrixtoks \\ }
        \repeat
      }
    }
  }
  \the\matrixtoks
}

\usetikzlibrary{trees}
\usetikzlibrary{decorations.pathmorphing}
\usetikzlibrary{decorations.markings}

```



```
\usetikzlibrary{
    positioning,
    arrows,
    decorations.markings,
    hobby
}

% tracer une ligne hachurée
\tikzstyle{mirror} = [
    draw,
    postaction=decorate,
    decoration={
        markings,
        mark=between positions 0.015 and 0.98 step 0.1072 with {\draw (0,0)--(60:7pt);}
    }
]

\tikzset{
    photon/.style={decorate, decoration={snake}, ->}
}

\tikzset{
    photon_par/.style n args={3}{
        decorate, decoration={snake, amplitude=#1, segment length=#2, post length=#3,
            pre length = #3/3}, ->}
}

\tikzset{
    resistance/.style n args={3}{
        decorate, decoration={zigzag, amplitude=#1, segment length=#2, post length=#3,
            pre length = #3}}
}

\tikzset{
    coil/.style n args={3}{
        decorate, decoration={coil, amplitude=#1, segment length=#2, post length=#3,
            pre length = #3, aspect = #2/14.28}}
}

%sets
\newcommand{\R}{\mathbb{R}}
\newcommand{\Z}{\mathbb{Z}}
\newcommand{\N}{\mathbb{N}}
\newcommand{\AN}{\mathbb{Z}^-}
\newcommand{\C}{\mathbb{C}}
\newcommand{\Q}{\mathbb{Q}}
```

```

\newcommand\numberthis{\addtocounter{equation}{1}\tag{\theequation}}
\newcommand\Ha{\mathcal{H}}
\newcommand\me{\mathrm{e}}
\DeclareDocumentCommand\Ldv{ s o m g d() }
{ % Total derivative
  % s: star for \flatfrac flat derivative
  % o: optional n for nth derivative
  % m: mandatory (x in df/dx)
  % g: optional (f in df/dx)
  % d: long-form d/dx(...)
  \IfBooleanTF{#1}
  {\let\fractype\flatfrac}
  {\let\fractype\frac}
  \IfNoValueTF{#4}
  {
    \IfNoValueTF{#5}
    {\fractype{\mathrm{D}} \IfNoValueTF{#2}{\^{#2}}{\mathrm{D}} #3\IfNoValueTF{#2}
      {\^{#2}}}}
    {\fractype{\mathrm{D}} \IfNoValueTF{#2}{\^{#2}}{\mathrm{D}} #3\IfNoValueTF{#2}
      {\^{#2}}}

    \argopen(#5\argclose)}
  }
  {\fractype{\mathrm{D}} \IfNoValueTF{#2}{\^{#2}} #3}{\mathrm{D}} #4\IfNoValueTF{#2}
    {\^{#2}}}
}

\newcommand{\qedi}{\quad_{\_}_{\qed}}

%EN TETE DEVOIR -----
\newcommand*\redacteur}[3]{\def\Nomredacteur{#1 \hfill #3\ \ #2}}
\newcommand*\coteCours}[1]{\def\CoteDuCours{#1}}
\newcommand*\nodev}[1]{\def\NoDev{#1}}
\newcommand*\doc}[1]{\def\Doc{#1}}
\newcommand\entete{
  \thispagestyle{firststyle}
  \flushleft
  \begin{minipage}{17cm}
  {\Large \Nomredacteur \par}
  \vspace{2mm}
  {\large \CoteDuCours}
  \end{minipage}

  \vspace{1cm}
  \headline{\textbf{{\Large \Doc \ , \NoDev}}}
  \vspace{1cm}
}

```

```
%Commande pour faire une en-tête
%\redacteur{NOM1}{NOM2}{JJ/MM/AA}
%\coteCours{CODE DU COURS}
%\titreDuCours{NOM DU COURS}
%\doc{devoir}
%\nodev{Numéros du devoir}
%\entete

%PAGE TITRE M1-----
\newcommand{\HRule}{\rule{\linewidth}{0.5mm}}
\newcommand{\blap}[1]{\vbox to Opt{#1\vss}}
\newcommand\AtUpperLeftCorner[3]{%
  \put(\LenToUnit{#1},\LenToUnit{\dimexpr\paperheight-#2}){\blap{#3}}%
}
\newcommand\AtUpperRightCorner[3]{%
  \put(\LenToUnit{\dimexpr\paperwidth-#1},\LenToUnit{\dimexpr\paperheight-#2})
  {\blap{\llap{#3}}}%
}

\newcommand*\redigeur[2]{\def\redigeur{#1 \#2}}
\newcommand*\correcteur[1]{\def\correcteur{#1}}
\newcommand*\information[3]{\def\info{#1 \#2 \#3 \vspace{2.5cm} #3}}
\newcommand*\Cote[1]{\def\cote{#1}}
\newcommand*\NomCours[1]{\def\NOM{#1}}
\newcommand\pageTitre{
  \thispagestyle{empty}
\begin{titlepage}
  \enlargethispage{2cm}

  \AddToShipoutPicture{
    \AtUpperLeftCorner{1.5 cm}{2.25cm}{\includegraphics[width=7cm]{logo.png}}
    \AtUpperRightCorner{1.5cm}{1cm}{\includegraphics[width=4cm]{UdeS.jpg}}
  }

  \begin{center}
    \vspace*{8cm}
    %\textsc{\NOM}\
    %\textsc{\cote}
    %\vspace*{3.5cm}

    \textsc{\@title}
    \HRule
    \vspace*{0.5cm}

    \large{\textsc{\redigeur}}
    \vspace{4cm}
```

```

        %Présenté à \
        %\textsc{\correcteur}
        %\vspace{1.5cm}
        \info

\end{center}

\vspace*{1.5cm}

\begin{center}
    \makebox[\textwidth]{\includegraphics[width=0.7\textwidth]{UdeS.png}}
\end{center}

\end{titlepage}
\ClearShipoutPicture
\makeatletter
\thispagestyle{firststyle2}
\cleardoublepage %oblige latex à 1. placer tout ce qu'il y a plus haut 2. à faire qu'en mo
}

%Commandes page titre après le \begin{document}:
% \title{\LARGE{TEMPLATE DE RAPPORT}}
% \redigreur{REDIGEUR 1}{REDIGEUR 2}
% \correcteur{CORRECTEUR}
% \Cote{cote du cour}
% \NomCours{Nom du cours}
% \information{Faculté des Sciences}{Département de physique}{\today}

%COULEUR PERSONNALISE-----
\definecolor{bleu}{RGB}{184,212,222}
\definecolor{gris}{RGB}{70,77,79}
\definecolor{VertCommantaire}{rgb}{0,0.6,0}
\definecolor{MauveChaine}{rgb}{0.58,0,0.82}
\definecolor{copper}{RGB}{170, 85, 0}
\definecolor{silver}{RGB}{200, 206, 210}
\definecolor{grey}{gray}{0.5}
\definecolor{dark grey}{gray}{0.2}
\definecolor{light grey}{gray}{0.8}
\definecolor{dark green}{RGB}{21,140,0}
\definecolor{light blue}{RGB}{145, 186, 255}

%INTRODUIT CODE -----

\newcounter{pythoncode} %Cré un nouveau compteur pour les blocs de pythoncode
\newtcblisting{pythoncode}[2][]{ %Permet de placer du code dans le texte pis que ce s
phantom={\stepcounter{pythoncode}}, %"phantom" execute un segmant de code. "\stepco
enhanced, %Permet de faire des choses plus cool

```

```
breakable,
title={Code \thepythoncode\hspace{20 mm}\textit{#2} },           %Le titre sera "Code " suivi
colframe=blue!50!black,           %Le comptour sera de cette couleur (bleu foncé)
colback=blue!10!white,           %Le derrière sera de cette couleur (bleu pale)
fonttitle=\ttfamily,           %La police d'écriture
coltitle=white,           %Couleur du titre (blanc)
%boxed title style={size=small, colback=blue!75!black},
width=\linewidth,           %La longueur de la boîte est de la longueur de la ligne
left=6mm,           %espacement entre le bord et les numéro de ligne (6mm)
listing only,
drop shadow,           %met de l'ombre derrière la boîte
listing options={           %Instructions transmises au truc qui place les lignes de code
  language=Python,           %Language de programmation
  basicstyle=\ttfamily\footnotesize,           %Le style de l'écriture (petit caractère)
  numbers =left,           %Afficher des numéros de lignes à gauche
  numberstyle=\color{gris},           %Couleur des numéros de ligne
  commentstyle=\color{VertCommantaire},           %Couleur des commantaires python
  inputencoding=utf8,           %Encodage utilisé
  keywordstyle=\color{blue},
  %identifierstyle=\color{orange},
  showspaces=false,           % show spaces everywhere adding particular underline
  showstringspaces=false,           % underline spaces within strings only
  showtabs=false,           % show tabs within strings adding particular underline
  stringstyle=\color{MauveChaine},
  escapechar=|,},           %Pour entré du code LaTeX entrer le caractère "|"
}
```