
TP2 – Rédaction informatique et mathématique – L1 · S1

plus loin en **LATEX**

2024

Objectifs

- manipuler les environnements de liste
- approfondir le mode math
- comprendre le mécanisme de compilation avec fichiers auxiliaires
- créer des tableaux

Exercice 1 *Échauffement*

1. Dans un nouveau fichier, écrivez le squelette d'un document **LATEX**, en utilisant la classe **article**, les paquets **inputenc**, **babel** et **fontenc** avec les options qui vont bien comme vus précédemment. Définissez un titre, un auteur et une date¹ dans le préambule, et faites-en sorte que ces informations soient affichées en haut de la première page.
2. Insérez dans ce document une liste numérotée, une liste non-numérotée (liste à puce) et une liste descriptive (comme vu en cours-td), avec, pour chacune, des éléments quelconques.
3. Insérez une (sous-)liste numérotée dans l'un des éléments de la liste numérotée créée précédemment, et ajoutez-y quelques éléments.
4. Dans chaque élément de la liste non-numérotée faites référence au deuxième élément de la liste numérotée. Combien de compilations devez-vous réaliser pour voir les références s'afficher correctement ?
5. Dans chaque élément de la liste descriptive faites référence au deuxième élément de la sous-liste numérotée, que vous avez créée à la question 3. Combien de compilations devez-vous réaliser pour voir les références s'afficher correctement ?
6. Placez les trois listes principales (celles créées à la question 2) dans trois sections différentes. Compilez. Les références changent-elles ?
7. Dans chaque élément de la liste numérotée, faites référence à la seconde section.
8. Ajoutez la table des matières en début de document.
9. Sauvegardez votre document en lieu sûr, nous nous en servirons à l'exercice 2.

1. La commande `\today` affiche la date d'aujourd'hui. Cependant, lorsque l'on déclare la date d'un document (méta-donnée) via `\date` dans le préambule, il est préférable d'indiquer la date explicitement pour que le fichier source soit daté (et non la compilation).

Exercice 2 *Mais que fait le bouton ► de texmaker ?*

1. Retrouvez votre fichier `.tex` de l'[exercice 1](#), et mémorisez le chemin complet du répertoire dans lequel il est stocké, puis fermez **texmaker** et ouvrez un terminal.

Nous allons étudier l'action provoquée par le bouton **►** de **texmaker**. Cela nous permettra également d'observer les fichiers auxiliaires et de se familiariser avec le terminal dans lequel se passera l'intégralité de cet exercice.

2. À l'aide de la commande `cd`², placez-vous dans le répertoire dans lequel le fichier `tex` de l'exercice précédent a été sauvegardé.
3. À l'aide de la commande `ls`³, listez le contenu du répertoire. Vous devriez y voir votre fichier. En trouvez-vous d'autres ? Quelles sont leurs extensions ?
4. À l'aide de la commande `rm -iv`⁴ , supprimez tous les fichiers du répertoire, à l'exception du précieux fichier `.tex`. Attention, si vous supprimez le fichier `.tex`, vous devrez recommencer l'[exercice 1](#).

Nous allons maintenant réaliser la compilation L^AT_EX depuis le terminal. Dans la suite de l'exercice, le nombre de compilations demandé est important. Si vous vous trompez, nettoyez votre répertoire en répétant la [question 4](#) puis reprenez à partir d'ici.

5. Listez les fichiers du répertoires. Seul le fichier `.tex` devrait apparaître.
6. À l'aide de la commande `pdflatex` compilez le document `.tex` une fois (il suffit de passer le nom du document, avec ou sans son extension, en argument).
7. Listez les fichiers du répertoires. Que constatez-vous ? Quels fichiers ont été créés depuis la [question 5](#) ?
8. Ouvrez le document `pdf`.⁵ Est-il conforme aux attentes (observez en particulier les références et la table des matières) ?
9. À l'aide de la commande `grep`, cherchez le mot '*Reference*' dans le fichier d'extension `.log`. Cherchez également le mot '*Label*' dans ce fichier. Que trouvez-vous et qu'est-ce que cela peut-il bien signifier ?
10. Compilez à nouveau. Y-a-t-il de nouveaux fichiers ? Le `pdf` a-t-il changé ?
11. Éditez le fichier `.toc` avec un éditeur de votre choix (`gedit` fera l'affaire), pour y inverser l'ordre des sections. Sauvegardez et fermez l'éditeur, puis compilez **une fois**. Voyez-vous les changements attendus dans le fichier `.pdf` ? Que remarquez-vous dans le fichier `.toc` ?

Le bouton **►** de **texmaker** effectue, dans l'ordre, les opérations suivantes : (1) sauvegarder le document source ; (2) compiler une fois le document à l'aide de `pdflatex` ; (3) signaler le changement au lecteur `pdf` pour qu'il rafraîchisse son affichage.

2. `cd` signifie *change directory*, ce qui en anglais signifie changer de répertoire.

3. `ls` signifie *list*, ce qui signifie lister.

4. `rm` signifie *remove*, soit supprimer en français. Les fichiers à supprimer sont passés en argument. Les options `-i` et `-v` sont conseillées.

5. Depuis le terminal, on ouvre un document `pdf` en utilisant la commande correspondant au lecteur `pdf` souhaité suivi du nom (avec extension) du document `pdf` à ouvrir, comme argument. Si on ne connaît pas le logiciel adapté à notre document, on peut utiliser la commande `xdg-open` qui essaie d'ouvrir un document (pas nécessairement un `pdf`) avec le logiciel par défaut pour ce type de document.

Exercice 3 Mathématiques discrètes

- Créez un document L^AT_EX en langue française, avec la classe `article`, les paquets : `inputenc`, `fontenc`, `babel`, `mathtools` et `amssymb` et les options qui vont bien. Donnez lui le titre “*Mathématiques discrètes*”, déclarez vous vous-même comme auteur, déclarez la date d’aujourd’hui¹, et faites en sorte que l’entête de titre s’affiche en haut de la première page. Sauvegardez. Votre document doit compiler.
- Dans le contenu du document, définissez l’ensemble des entiers naturels comme l’ensemble des entiers positifs ou nuls. Introduisez la notation \mathbb{N} pour cet ensemble, que vous obtiendrez, en mode math, par `\mathbb{N}`.

Dans la suite du document nous allons écrire des théorèmes et des preuves. L’environnement `theorem` n’est pas défini par défaut dans la classe `article`, nous allons donc le définir. Le paquet `amsthm` (de l’*American Mathematical Society*) fournit des commandes (en particulier la commande `\newtheorem`⁶) pour aider la création des environnements sémantiques pour les maths (théorème, proposition, lemme...) ainsi qu’une définition de l’environnement `proof` (preuve) qui insère automatiquement le symbole `\qed` de fin de preuve à la fin.

- Dans le préambule, importez le paquet `amsthm`, puis définissez l’environnement `theorem` par la commande suivante :

```
\newtheorem{theorem}{Théorème}
```

Ici, le premier argument (`theorem`) est le nom de l’environnement à créer (on fera donc `\begin{theorem}... \end{theorem}`) et le second argument est le titre qui sera affiché dans le pdf (voir question suivante).

- Créez une section, que vous nommerez “*Somme des premiers entiers naturels*” puis écrivez le théorème suivant⁷ :

Théorème 1. *Pour tout entier naturel n , la somme des n premiers entiers naturels est égal à la moitié du produit de n par $(n - 1)$.*

(Remarquez que le premier entier naturel est 0.)

- Recopiez⁸ le code L^AT_EX donné en Listing 1 ; compilez ; observez.
- Dans le code copié à la question précédente :
 - À quoi sert l’environnement `align*` et, à l’intérieur, l’esperluette (`&`) et `\backslash` ?
 - Dans quel mode est-on à l’intérieur de l’environnement `align*` ?
 - Que fait `\times` ?
 - À quoi sert la commande `\text` ?
 - Quelle différence y-a-t-il entre les commandes `\ldots` et `\cdots` ?
 - À quoi servent les commandes `\left` et `\right` qui fonctionnent par paire ?

6. Techniquement parlant, le paquet améliore la commande `\newtheorem` qui existe en L^AT_EX (essayez).

7. Le cadre ne doit pas être affiché, il sert, ici et dans l’ensemble de ce sujet de TP, à vous montrer un extrait du rendu attendu.

8. Le copier-coller depuis le pdf est déconseillé, car, suivant le lecteur pdf utilisé, il peut être erroné (introduction d’espaces, substitutions de caractère...).

Listing 1 – Début de preuve du théorème de la [question 4](#) pour la [question 5](#).

```
\begin{proof}
Soit  $n \in \mathbb{N}$ .
Notons  $S_n$  la somme des  $n$  premiers entiers naturels.
Alors:
\begin{align*}
S_n &= \sum_{i=0}^{n-1} i \\
&= 0 + 1 + \cdots + (n-2) + (n-1) \\
&\quad & \text{par définition} \\
&\quad & \& \\
&\quad & = \sum_{i=0}^{n-1} (n-1-i) \\
&\quad & = (n-1) + (n-2) + \cdots + 1 + 0 \\
&\quad & & \text{par commutativité} \\
\end{align*}
En sommant les deux formes de la somme  $S_n$  données ci-dessus,
nous pouvons obtenir:
\begin{align*}
2 \times S_n
&= \left( \sum_{i=0}^{n-1} i \right) \\
&\quad + \left( \sum_{i=0}^{n-1} (n-1-i) \right) \\
&\quad & \text{par associativité et commutativité} \\
&\quad & \& \\
&\quad & = \text{\texttt{\text{\\text{\\emph{\\ldots<complétez la preuve ici>\\ldots}}}}} \\
\end{align*}
\text{\texttt{\text{\\emph{\\ldots<concluez la preuve ici>\\ldots}}}}
\end{proof}
```

7. Complétez et concluez la preuve de la [question précédente](#).⁹
8. Utilisez l'argument optionnel accepté par l'environnement `theorem` pour indiquer “*version textuelle*” comme détail au théorème de la [question 4](#), donnant ainsi :

Théorème 1 (version textuelle). *Pour tout entier naturel n , la somme des n premiers entiers naturels est égal à la moitié du produit de n par $(n - 1)$.*

9. Donnez une seconde version du théorème (dans un second environnement `theorem`) avec le détail “*version mathématique*”, et écrite entièrement en symbole mathématique dans un environnement `equation*`. Vous pouvez utiliser le symbole `\forall` et la commande d'espacement `\colon` qui fonctionne en mode math. Faites en sorte que la valeur de la somme s'écrive sous cette forme : $\frac{n(n - 1)}{2}$.
10. Donnez une preuve par récurrence.
11. La somme des n premiers entiers impairs ($1 + 3 + \cdots + (2n - 1)$) vaut n^2 . Dans une seconde section, écrivez un théorème énonçant cette vérité, puis prouvez le, en utilisant le résultat précédent auquel vous ferez **référence**.

9. Un des objectifs de cette UE est de vous faire travailler votre raisonnement mathématique et votre aptitude à rédiger correctement des preuves. Vous retrouverez donc régulièrement des preuves à rédiger, sur des problèmes simples comme celui-ci.

- Que vaut la somme des n premiers entiers pairs ? Racontez tout cela dans une troisième section.

Exercice 4 Quantificateurs en tableau

- Dans l'éditeur (`gedit`? `vim`? `emacs`?) ou l'IDE (`texmaker`?) de votre choix, créez un nouveau fichier `tex`, avec la classe `article`, les paquets `inputenc`, `fontenc`, `babel`, `mathtools` et `amssymb` et les options qui vont bien. Donnez lui le titre “*Quantificateurs en tableau*”, déclarez vous vous-même comme auteur, déclarez la date d’aujourd’hui¹⁰, et faites en sorte que l’entête de titre (titre, auteur, date) s’affiche en haut de la première page. Sauvegardez ; compilez.

Nous allons créer des tableaux, à l'aide de l'environnement `tabular` qui attend un argument obligatoire. Cet argument spécifie la liste des colonnes (et donc leur nombre), ainsi que l'alignement du texte à l'intérieur, ainsi :

- l** aligné à gauche (*left*)
- c** centré (*center*)
- r** aligné à droite (*right*)
- p** justifié (*paragraph*), mais dans ce cas, il faut indiquer une largeur de colonne à LATEX qui en a besoin (e.g., `p{2cm}`) – oubliions ceci pour ce TP.

Par exemple, pour faire un tableau à deux colonnes, la première alignée à droite et la seconde alignée à gauche, on initiera l'environnement par `\begin{tabular}{rl}`. À l'intérieur de l'environnement, le contenu du tableau est donné ligne par ligne. Les cellules d'une même ligne sont séparées par le caractère spécial `&`, tandis que les lignes sont séparées par la commande `\backslash\backslash`.¹⁰

- Créez un tableau à deux colonnes. Dans la première colonne, écrivez, sur des lignes différentes, les symboles \forall (`\forallall`) et \exists (`\existsists`), et dans la seconde colonne, donnez la traduction du symbole correspondant en français. Compilez. Le résultat est-il conforme à vos attentes ?
- Ajoutez des titres au colonnes (e.g., “**quantificateur**” et “**signification**”). Mettez ces titres en gras, en les passant en argument à la commande `\textbf`.
- Pour tracer des bordures horizontales entre les lignes du tableau, il suffit de placer la commande `\hline` en début de ligne (au début de l'environnement, pour la première ligne, ou juste après `\backslash\backslash` pour les lignes suivantes). Séparez la ligne de titre des autres avec une bordure horizontale.
- Pour tracer des bordures verticales, il faut placer le caractères `|` autour des spécifications de colonne, dans l'argument de l'environnement `tabular`. Placez des bordures verticales et horizontales autour du tableau, mais pas entre les colonnes.
- Tracez les bordures (verticales et horizontales) entre cellules, mais doublez¹¹ la bordure horizontale séparant la ligne de titre de colonne des autres lignes.

10. Vous pouvez trouver une documentation plus complète [ici](#).

11. Les bordures verticales devraient présenter de petites interruptions, tout à fait appréciables.

Exercice 5 Vous avez dit « tableau » ?

En typographie, les guillemets ouverts sont différents des guillemets fermés (*e.g.*, “*c'est moche comme ça*” / “*c'est mieux comme ceci*”). De plus, les symboles utilisés sont différents selon la langue. Par exemple, en français, les guillemets s'écrivent habituellement «*hey*», tandis qu'en anglais, ils s'écrivent ‘*hey*’ (anglais UK) ou “*hey*” (anglais US). Puisque l'encodage de notre fichier source est l'`utf8` (par l'utilisation de l'encodage par défaut dans les versions récentes de L^AT_EX, ou par l'import du paquet `inputenc` avec l'option `utf8` dans les versions plus anciennes), nous pouvons directement écrire le symbole souhaité dans le code source. Cependant, puisque ces symboles ne sont pas toujours simplement accessibles, et surtout puisque d'autres encodages sont possibles, il existe des commandes L^AT_EX pour les réaliser. Tableau 1 donne des exemples.

TABLE 1 – Tableau de guillemets

nom	nom anglais	utf8	code L ^A T _E X	rendu
simples guillemets	<i>simple quotes</i>	‘hey’	‘hey’	‘hey’
doubles guillemets	<i>double quotes</i>	“hey”	“hey”	“hey”
guillemets français	<i>french quotes</i>	«hey»	\og hey\fg	« hey »

1. À votre avis, que signifient les sigles `og` et `fg` dans les commandes `\og` et `\fg` qui permettent d'écrire les guillemets français ? Ces commandes sont-elles définies dans un document qui n'est pas en français (*i.e.*, sans l'option `french` à l'import de la classe, ni à `babel`) ?
2. Recopiez le tableau 1. Vous prendrez soin aux points suivants :
 - a) la ligne de titre est séparée des autres lignes par une bordure horizontale simple ;
 - b) dans chaque colonne, l'alignement du texte est centré ;
 - c) dans la colonne titrée `utf8`, vous ne devez utiliser aucune commande (si ce n'est `\textbf` pour mettre le titre de la colonne en gras) mais vous devez entrer directement le caractère `utf8` (vous avez le droit au copier-coller) ;
 - d) vous devez utiliser la commande `\verb12` pour afficher le code source dans la colonne titrée `code LATEX` ;
 - e) en *verbatim*, les symboles de guillemets apparaissent comme sur le sujet – pour cela, vous devriez avoir besoin d'importer le paquet `upquote` ;
 - f) L^AT_EX s'écrit L^AT_EX et pas LaTe_X ;
 - g) le contenu de la colonne de rendu doit être égal à celui de la colonne de code L^AT_EX, sans la commande `\verb`.

12. Cf. présentation du premier cours-td.