Plan Régional de Formation
M. Krieg - M. Schavsinski

Initiation à LaTeX- Bien débuter

I) Le préambule

1) Qu'est-ce que c'est ?

Un document LaTeXest constitué de deux grandes parties :

- le préambule
- le "corps"du document

Dans le préambule, nous allons appeler tous les environnements, toutes les bibliothèques, définir toutes les commandes, créer tous les raccourcis dont nous avons besoin pour taper notre document.
Le préambule va donner la forme du document et est donc un élément personnel en fonction de vos besoins et de votre manière de coder.
On peut y définir les règles de mise en page que l'on souhaite voir appliquer à l'ensemble du document : ainsi, si on veut changer la mise en page, la police ou autre chose, il suffit de changer les éléments dans le préambule, puis de compiler à nouveau : on n'a pas besoin de parcourir tout le document pour changer des éléments partout.
On peut remarquer par exemple que, dans ce document, un nouveau paragraphe ne commence pas par un alinéa : c'est l'un des éléments définis dans le préambule.
Le préambule doit toujours commencer par la classe du document que vous tapez et il se termine au début du corps du document, c'est-à-dire au \begin{document}

2) Quels sont les éléments invoqués dans le préambule

Il sont de quatre natures :

1. La classe de document : c'est le premier élément qui intervient (et il n'intervient qu'une fois).
   La classe est définie par la commande : `\documentclass[11pt,letterpaper]{article}`.
   Entre les accolades, la classe de votre document (ici article) et entre crochets, les options.
2. Les appels à des "packages". Les packages sont des ensembles de commandes prédéfinies. Quand on va faire appel à ces commandes, il faut préalablement avoir invoqué les packages correspondants.
   L'invocation se fait ainsi : `\usepackage[utf8]{inputenc}`
   Là encore, la commande usepackage fait appel au package inputenc, avec l'option utf8.
3. Les définitions de commandes personnelles, ou redéfinitions.
   On peut créer une commande personnelle, au moyen de la commande `\newcommand{\R}{\mathbb{R}}`.
   Cette commande prend deux arguments : le premier argument (entre la première paire d'accolades) est le nom de la commande que vous créez, c'est à dire ce qu'il faudra taper dans le corps du document pour utiliser cette commande, le second argument est la série d'instructions que cette commande doit déclencher.
   Dans l'exemple donné ici, on créée une commande `\R`, dans le corps du document, taper `\R` sera l'équivalent de taper `\mathbb{R}`. C'est donc un raccourci pour une instruction ou série d'instructions que l'on va utiliser avec une certaine régularité. Ici, c'est pour créer un symbole $\mathbb{R}$ dans le mode mathématique (la commande `\mathbb{...}` affiche le texte entre les accollades avec la police des lettres ajourées que l'on écrit au tableau noir (Math Black Board)).
   Pour redéfinir une commande déjà définie, on utilise le même principe, mais avec la commande :
   `\renewcommand{...}{\...}`. Pour savoir si une commande était déjà définie, on fait comme si elle ne l'était pas (avec newcommand, donc) et la compilation renverra un message d'erreur :
   ligne 109 : Command \R already defined. \newcommand{\R}{...})
   cela indique que la commande que l'on pensait définir à la ligne 109 a un nom qui est en conflit avec une commande déjà existante : soit on change le nom de la commande définie ligne 109, soit toujours à la ligne 109, on passe de l'instruction newcommand à renewcommand.
4. Des définitions d'options, de longueurs, d'éléments de métacontenu (nom de l'auteur, contenu des en-têtes et pieds de pages...)

Bref, avec tous ces éléments, le préambule peut être assez long (le préambule type des documents de l'APMEP est de plus d'une soixantaine de lignes).

3) Choix de la classe

il existe plusieurs classes prédéfinies dans LaTeX :

1. article : pour des articles destinés à la publication et ne contenant que quelques pages ;
2. report : pour des documents un peu plus longs contenant plusieurs chapitres, comme des mémoires de thèse ;
3. book : pour de véritables livres, de plusieurs centaines de pages.
4. slides : pour faire des présentations sur transparents.
5. beamer : pour faire des présentations utilisant la magnifique extension beamer ;
6. lettre : pour faire des lettres au format français (classe écrite par l'Observatoire de Genève).
7. memoir : pour écrire des mémoires, par exemple de fin d'étude.

(liste et description tirées de Wikipédia)

Concrètement, au quotidien, vous utiliserez sans doute essentiellement article et slides ou beamer (voire seulement article).

4) Classe article

C'est celle qu'on utilise dès que l'on veut faire un document de quelques pages, avec des pages dans un format classique.
Les principales options de la classe article (elles peuvent être mentionnées dans un ordre quelconque) :

1. `10pt`, `11pt` ou `12pt` : choix de la taille de la police pour le corps de texte de votre document.
2. `letterpaper`, `a4paper` ... : choix de la dimension de la page
3. `landscape` : orientation paysage (l'orientation portrait est par défaut).
4. `twocolumn` : présentation de l'intégralité du document en deux colonnes.
5. `twoside` : précise que le document doit être imprimé en recto-verso, ce qui permet de traiter les marges différemment pour les pages paires et les pages impaires.

5) Un exemple de préambule

Nous vous proposons d'utiliser le préambule fourni sur le site de la formation Interface sur Prefas afin de pouvoir vous concentrer sur l'essentiel pour l'instant : **Ecrire du texte** et des maths si possible ;)
Vous pourrez aller voir à la fin de ce document pour plus de détails sur les éléments de ce préambule.
Vous devez donc copier le contenu de "monpreambule" pour le mettre au début de votre document.

II) taper du texte

1) Hello World!

Comme dans n'importe quel language de programmation (même si LaTeXn'en est pas un), nous allons commencer par écrire le traditionnel Hello World!
Nous allons donc commencer notre document à la suite du préambule par : \begin{document}
et nous le terminerons par \end{document}. Tout ce qui sera écrit après cela ne sera pas lu par notre interpréteur.
Votre texte s'écrit donc :
\begin{document}
Hello World!
\end{document}
Enregistrons notre document en allant dans Fichiers puis Enregistrer sous.
Nous vous conseillons de le placer dans un dossier créé pour l'occasion.
Pour observer le résultat, nous allons cliquer sur la flèche à côté de compilation rapide dans TexMaker.

En fonction de la taille de votre écran, vous voyez apparaître le rendu de votre document à droite de la fenêtre de code ou par dessus.
Ce document est le document pdf final, même si LaTeXa créé une série de fichiers pour y parvenir (fichier journal, fichier de compilation, etc ...) qui ne vous servent plus et qui pourront être jetés, LaTeXles recréera si vous recompilez.
C'est la raison pour laquelle il faut placer vos documents dans des dossiers créés pour l'occasion car le nombre de fichiers explose rapidement et il vaut mieux s'éviter la séance de ménage du bureau qui s'en suivrait.

2) Taille du texte

La taille de base du texte est définie tout au début du code (et les trois seules options sont 10 pt, 11 pt et 12 pt). Cette taille est la taille du texte dans les paragraphes. Le texte dans les titres de section et autres, est de taille différente, mais proportionnelle à la taille de base définie.
Quand, ponctuellement, on veut changer la taille d'une portion de texte, on peut le faire au moyen de commandes qui indiquent la taille :

1. {\tiny …} tout petit
2. {\scriptsize …} taille des indices
3. {\footnotesize …} taille des notes de bas de page
4. {\small …} petit
5. {\normalsize …} taille de base
6. {\large …} grand
7. {\Large …} très grand
8. {\LARGE …} très très grand
9. {\huge …} énorme
10. {\Huge …} vraiment énorme

3) Format du texte

On peut changer l'apparence du texte en utilisant diverses commandes :
— **mettre en gras** : avec \textbf{…}. Le texte à mettre en gras est à insérer entre les accolades.
Explication de la commande : text car la commande s'utilise en mode texte et bf pour Bold Face.
— *mettre en italique* : avec \textit{…}.
— *mettre en* évidence *un texte* : avec la commande \emph{…}. Distinction subtile : la mise en évidence d'un texte bascule le texte à mettre en évidence en italique, si le texte de base est en romain, et en romain si le texte de base est en italique.
emph pour emphasise : mettre en évidence.
— souligner du texte : avec la commande \underline{…}. Les puristes de la typographie considèrent que ce n'est pas joli, et que c'est à éviter (mais on a le droit de ne pas être puriste).

4) Paragraphes

Pour taper du texte, pas de difficultés particulières, vous n'avez qu'à taper le texte normalement. Attention, cependant, quand vous passez à la ligne dans le code LaTeX, cela ne provoque pas un passage à la ligne dans le texte compilé. Pour forcer un retour à la ligne, plusieurs solutions : vous pouvez laisser une ligne blanche dans votre code, cela sera compilé comme un nouveau paragraphe.

C'est ce que j'ai fait ici comme depuis le début du document et que je n'ai pas fait au paragraphe précédent, et c'est pour cela que ce paragraphe commence avec une indentation (alinéa). On remarquera que par défaut, le premier paragraphe d'une section n'est pas indenté. On verra plus tard comment gérer cela.

Une autre façon de de créer un nouveau retour à la ligne, c'est d'utiliser la commande \par. Cela créera un nouveau paragraphe, sans ligne blanche dans le code.
Une troisième façon de créer un retour à la ligne, mais sans créer de nouveau paragraphe, c'est d'utiliser la commande \\. On remarquera ici que j'ai créé un retour à la ligne, mais que je reste dans le même paragraphe, car il n'y a pas d'indentation.

5) Alignement

Par défaut le texte est justifié (sauf pour les lignes terminant un paragraphe), et LaTeX inclus des fichiers qui indiquent à quel endroit effectuer la césure d'un mot en fonction de la langue utilisée (indubitablement, c'est très sophistiqué. J'ai tapé indubitablement, c'est un mot long qui a été découpé automatiquement ici).

On peut centrer le texte, avec l'environnement \begin{center} …\end{center}.
Les environnements flushleft et flushright permettent d'aligner à gauche et à droite respectivement.

6) Le saut de ligne

Le saut de ligne peut se faire par plusieurs méthodes :
— On utilise les commandes \miniskip, \medskip ou \bigskip qui laissent des espaces plus ou moins grands
— une commande moins esthétique $ $ qui utilise un artifice de commande mathématique vide
— pour un saut plus important, on peut utiliser \vspace{*taille du saut* cm }

III) On Tape des mathématiques ?

1) Les deux modes mathématiques

Pour taper des mathématiques, il y a deux modes : le mode "hors ligne centré", qui permet d'obtenir quelque chose dans ce genre :

$$\int_0^{+\infty} f(t)\mathrm{d}t$$

et le mode "en ligne", qui avec le même code, va afficher : $\int_0^{+\infty} f(t)\mathrm{d}t$.

Dans un article scientifique, on préfère utiliser le mode hors ligne centré, c'est la norme, mais dans un document au quotidien, c'est aussi consommateur de place et de papier, mais le mode en ligne donne des choses moins jolies.

On peut choisir quelque chose d'intermédiaire, en affichant des formules compilées comme en mode hors ligne centré, mais sans avoir le passage à la ligne et le centrage du texte : cela donne $\displaystyle\int_0^{+\infty} f(t)\mathrm{d}t$.

Le mode mathématique hors ligne centré est délimité par \[...\].

Le mode mathématique en ligne est délimité par $ ...$.

Pour forcer l'affichage de type hors ligne centré dans le mode en ligne, on utilise la commande \displaystyle au début.

2) Le romain en mode mathématique

En mode mathématique, par défaut toute lettre est considérée comme une variable, et est donc affichée en italique, ce qu'il faudra parfois éviter (comme pour le d de dt dans mon exemple précédent). Pour cela, on précisera que l'élément est à écrire en romain via la commande \text{...}.

Cependant, pour un certain nombre d'occasions, il y a plus court : notamment, toutes les fonctions usuelles (sin, cos, tan, ln, exp...) doivent être composées en romain. Pour ces fonctions, on obtient ce résultat en précédant le nom de la fonction d'un antislash : $ f(x)= -3\ln x $ donne $f(x) = -3\ln x$.

En mode mathématique :

— le f(x) sera composé avec f et x en italique, mais les parenthèses en romain.

— les espaces dans le code seront ignorées (on peut voir que dans mon code, je n'ai pas mis d'espace avant le = et j'en ai mis une après, mais quand le code est interprété, mon signe = est entouré de chaque côté d'une espace).

— le tiret (-) est interprété comme un signe moins (−) (plus long, et verticalement centré par rapport au égal).

3) Les commandes en mathématiques

— indice : le texte à mettre en indice est indiqué par le code suivant : _{...}. S'il n'y a qu'une lettre à mettre en indice, on peut la placer après le tiret bas (underscore) sans forcément la mettre entre accolades.

— exposant : même principe, mais avec le code : ^{...}.

— symboles divers :

  — $\neq$ \neq (**n**ot **eq**ual to).

  — $\equiv$ \equiv.

  — $\times$ \times .

  — $\div$ \div.

  — $\leq$ \leq (**l**ess than or **eq**ual to), ou bien, plus français : $\leqslant$ \leqslant (slanted veut dire penché), et aussi $\geq$ et $\geqslant$ obtenus avec \geq et \geqslant.

  — ∞ \infty.

  — $\pm$ \pm et son jumeau maléfique : $\mp$ \mp.

  — $\approx$ \approx et $\simeq$ \simeq (*a priori* à réserver pour asymptotiquement équivalent).

  — ∂ \partial.

  — $\cap$ \cap et $\cup$ \cup.

  — $\in$ \in ; $\notin$ \notin et $\subset$ \subset ; $\not\subset$ \not\subset.

  — $\forall$ \forall et $\exists$ \exists.

  — $\perp$ \perp.

  — $\cdot$ \cdot (point de multiplication, centré verticalement sur le signe égal ou le signe moins).

  — $\circ$ \circ (composition)

  — La fiche wikipédia intitulée « Table de symboles mathématiques » donne une liste plus complète.

— Opérateurs (version normale, puis displaystyle) :

  — $\int_a^b$ ou $\displaystyle\int_a^b$ \int_{a}^{b}.

  — $\sum_a^b$ ou $\displaystyle\sum_a^b$ \sum_{a}^{b}.

- $\prod_a^b$ ou $\prod_a^b$ \prod_{a}^{b}.
- $\lim_a$ ou $\lim_a$ \lim.
  Pour cet opérateur et les trois précédents, on peut obtenir le mode displaystyle en intercalant \limits juste après le nom de l'opérateur.
- $\frac{a}{b}$ ou $\frac{a}{b}$ \frac{a}{b}.
  on peut obtenir l'affichage displaystyle en tapant \dfrac plutôt que \displaystyle\frac.
- $\sqrt{a}$ \sqrt{a} et $\sqrt[n]{a}$ \sqrt[n]{a}

— chapeaux et autres
- $\vec{a}$ \vec{a}.
- $\hat{a}$ \hat{a} et $\widehat{ABC}$ \widehat{\text{ABC}}. Marche aussi avec tilde et widetilde...

— Délimiteurs on peut utiliser () [] etc. Pour afficher des accolades, il faut les précéder d'un antislash.
Si on veut des délimiteurs qui s'ajustent en hauteur au contenu qu'ils enserrent, il faut utiliser les commandes \left et \right suivies du symbole à représenter. Ces deux commandes doivent toujours être utilisées. Si on veut un délimiteur d'un côté mais pas de l'autre (pour un système d'équations, par exemple) on met quand même les deux, et on affiche un . après celui qui ne doit pas apparaître.
On peut utiliser à droite un délimiteur différent de celui qui est à gauche. on a le choix entre :
( ) { } [ ] ⟨ ⟩ | ‖
Exemple : $\left]\left\{\sqrt{3}\right)[t]\int_a^b f(t)\mathrm{d}t$ est obtenu avec le code :
$\left]\left\{\sqrt{3} \right)[t] \int\limits_{a}^{b} f(t) \text{d}t \right.$

— Flèches :
- $\leftrightarrow$ \leftrightarrow et $\Leftrightarrow$ \Leftrightarrow
- $\longleftrightarrow$ \longleftrightarrow et $\Longleftrightarrow$ \Longleftrightarrow ou \iff
- $\mapsto$ \mapsto et $\longmapsto$ \longmapsto
- …

— lettres grecques, en mode mathématique : δ \delta et Δ \Delta...
Pour certaines, il existe des variantes : ϵ \epsilon ou ε \varepsilon, ϕ \phi ou φ \varphi, à vous d'explorer.

IV) Organiser le document

On peut créer des \section ; des \subsection ; des \subsubsection etc. La numérotation est automatique, et crée une structure dans le document (qui par la suite permet la création de tables des matières automatiques).
On peut créer des listes numérotées ou non numérotées. Le principe est simple : on crée un "environnement" de liste, avec \begin{itemize} et \end{itemize} (pour une liste à puce) \begin{enumerate} et \end{enumerate} (pour une liste numérotée).
Chaque élément de la liste (qu'elle soit numérotée ou à puce) sera identifié par la commande \item

V) C'est à vous!!!

Prenez un de vos devoirs et tapez-le en LaTeX.
Vous pouvez copier coller votre devoir en venant de tout éditeur, il va falloir retravailler beaucoup d'éléments mais c'est la meilleure manière de comprendre les changements par rapport à vos habitudes.

VI) Explications du préambule

1) Les packages

Rappel : Pour invoquer un package, la commande c'est : `\usepackage[options]{nom du package}`

a) Les indispensables

1. `fontenc` utilisé avec l'option `[T1]` : permet d'indiquer que quand vous saisissez un accent au clavier, cet accent sera reconnu et interprété correctement.
2. `inputenc` utilisé avec l'option `[utf8]` : permet d'indiquer que le document produit doit inclure les accents.
3. `amsmath, amsfont, amssymb` : deux packages de l'American Mathematical Society, qui regroupent de nombreuses commandes pour taper des mathématiques, et des symboles. Parfois avec des conventions de notation nord-américaines, qui font que certaines mises règles typographiques ne sont pas parfaitement adaptées aux conventions typographiques françaises.

4. `babel`, avec l'option `[french]` : le package babel permet d'utiliser LaTeX dans différentes langues. l'option french (précédemment nommée frenchb, ou francais) permet de dire que l'on va écrire en français, et donc de permettre à LaTeX de gérer correctement les règles d'espacement avant et après les signes de ponctuation, et les césures de mots en bout de ligne (quand c'est nécessaire).
   Si votre document est plurilingue, vous listez les langues que vous utilisez, séparées par des virgules. Celle qui est mentionnée en dernier sera active au début du document, ce sera la langue principale du document.
   Pour changer de langue : `\begin{otherlanguage}{greek} alphabet \end{otherlanguage}` donne :
   On notera que ce n'est pas le même rendu que d'utiliser $\alpha\lambda\beta$... en mode mathématiques (lettres en italique en mode mathématiques, espacement différent ...).
   Remarque : si votre document est en anglais, l'option `english` fait référence à l'anglais US, c'est l'option `british` pour l'anglais britannique.
   La commande `\{frenchbsetup}`, à placer dans le préambule de chaque document après le chargement de Babel, permet de personnaliser le comportement de babel-french grâce au large choix parmi des options disponibles.
   On peut citer `StandardLists=true`, qui renvoie des listes à mise en forme américaine (avec notamment, la puce de base pour les liste non numérotées qui est un gros point, plutôt qu'un tiret qui peut être confondu avec un signe de soustraction).
   Une liste plus complète peut être trouvée à l'adresse suivante : http ://daniel.flipo.free.fr/frenchb/frenchb-doc.pdf
5. `geometry`, package capital pour gérer la géométrie de la page (marges, position des en-têtes et pieds de page...).
   C'est possible de faire sans, mais c'est tellement plus compliqué...
   Les options du package geometry sont nombreuses. On peut commencer par y mettre les même options que celles utilisées dans la déclaration de classe, mais ce n'est pas la peine de le faire de façon redondante.
   Beaucoup plus intéressant, on peut définir ses marges. En effet, dans la classe article, les marges sont calculées en fonction d'un principe (ou d'un axiome) disant qu'une ligne est plus lisible quand elle contient un nombre limité de signes, et donc pour que les lignes ne soient pas trop pleines, les marges sont énormes. Dans geometry, on peut définir simplement les marges souhaitées.
   Des options possibles sont :
   — `[margin = 1cm]` : toutes les marges sont définies à 1 cm.
   — `[hmargin = 1in]` : les marges horizontales sont définie à 1 pouce.
   — `[vmargin = 14mm]` : les marges verticales sont définies à 14 mm.
   — `[left = 1cm, right = 2cm, top=3cm, bottom = 4cm]` : ... de façon assez transparente. Notez que vous pouvez combiner les options, en les enchaînant les unes à la suite des autres.
   left ou inner désigne la marge de gauche si le document est `oneside`, la marge intérieure (pour la reliure) si le document est `twoside` et de façon analogue pour right ou outer.
   — Pour les en-tête (header) ou pied de page (footer), on a plusieurs options : par défaut, ils sont placés dans la marge (haute ou basse). Si vous voulez que la dimension annoncée pour votre marge soit respectée et que l'en-tête ou le pied de page soit à cette distance du bord, il faut utiliser l'option `includehead` ou `includefoot` ou les deux (ce qui est équivalent à l'option `includeheadfoot`).
   `headheight` définit la hauteur de votre en-tête, `headsep` définit la séparation entre l'en-tête et le texte. on obtient les équivalents pied de page en remplaçant head par foot.
   — `columnsep = 1cm` : définit une séparation entre les colonnes de 1 cm.

b) Les optionnels

1. des packages de police `fourier`, `[scaled=0.875]helvet`, ou de symboles `wasysym`.
2. fancybox : pour faire de jolies "boîtes", c'est à dire des cadres.
3. `tabularx` : pour faire des tableaux (plus de détails plus tard)
4. `mathrsfs` : pour avoir des caractères en script en mode mathématiques $\mathbb{C}$, que l'on invoque avec la commande : `\mathbb{C}`.
   C'est une alternative à la commande `\mathcal{C}` qui donne $\mathscr{C}$: c'est une affaire de goût.
5. `fancyhdr` : (fancy header) pour faire de jolis en-têtes ou pieds de page.
6. `enumitem` : permet de personnaliser les listes numérotées et listes à puce.
7. `lscape` : permet de basculer à une mise portait à paysage (ou réciproquement) en cours de document.
8. `sectsty` : permet de mettre en forme les section (éléments structurants du document)
9. `titlesec` : un peu la même idée que le précédent.

2) newcommand

Voici des commandes newcommand interessantes pour les matheux :

1. pour les ensembles de nombres
2. pour les vecteurs
3. pour les propriétés, définitions, théorèmes, dans leur mise en page