

---

# TP1 – Rédaction informatique et mathématique – L1 · S1

## premiers pas avec $\text{\LaTeX}$

SEPTEMBRE 2025

---

### Objectifs

- maîtrise de l’environnement  $\text{\LaTeX}$  proposé sur les machines du SCI
- savoir écrire un document  $\text{\LaTeX}$  minimal
- savoir compiler un document  $\text{\LaTeX}$
- maîtriser le mécanisme de référencement interne de  $\text{\LaTeX}$
- découvrir les *packages* et les commandes  $\text{\LaTeX}$  de bases

### Instructions de démarrage

Connectez-vous sur la machine virtuelle “Latex Basique” (que vous pouvez retrouver chez vous sur <https://scep.prox.dsi.uca.fr/>). Attention, votre machine virtuelle n’est pas persistente : pensez à sauvegarder les fichiers que vous souhaitez conserver à la fin de votre session de travail. Il est en revanche déconseillé de travailler directement sur le disque `M_USER`, qui est distant donc lent et fragile (problèmes temporaires de synchronisation possibles) : travaillez sur la machine, puis copiez sur `M_USER` les fichiers qui en valent la peine.

Lancez le logiciel `texmaker`.

### Exercice 1 Hello World!

Nous allons créer un document  $\text{\LaTeX}$  minimaliste, composé d’un préambule minimaliste (juste la déclaration de classe qui est obligatoire), et d’un contenu minimaliste (une phrase).

1. Depuis `texmaker`, créez un nouveau document (vide).
2. Déclarez la classe du document au début du document : nous choisirons la classe `minimal`<sup>1</sup>, dont on passera le nom (`minimal`) en argument de la commande `\documentclass`, comme vu en cours-td.
3. Écrivez les deux commandes d’environnement, définissant la partie contenu du document (*c.f.* cours-td).
4. À l’intérieur de cet environnement, écrivez le texte “*Hello World!*” (sans les guillemets).
5. Depuis `texmaker`, tentez de compiler votre document, (utilisez le premier bouton ► précédé de l’action qu’il déclenche). Que se passe-t-il ?<sup>2</sup>

---

1. La classe `minimal` n’est en pratique pas utilisée : elle sert généralement de base pour définir des classes plus complexes – ce qui ne sera pas vu dans ce cours.

2. Dans cette question, le mot « tentez » et l’interrogation finale, devrait vous faire comprendre qu’un problème ne serait pas si surprenant que ça. Voyez puis passez à la question suivante. . .

6. Enregistrez votre fichier<sup>3</sup> sous un nom adéquat. L’extension des fichiers L<sup>A</sup>T<sub>E</sub>X est `.tex` (mais si vous ne spécifiez pas d’extension, elle devrait être ajoutée automatiquement par `texmaker`).
7. Compilez votre document ; visualisez le résultat.

## Exercice 2 *Bonjour monde !*

1. Ouvrez le document de l’exercice 1 dans `texmaker` et sauvegardez-le sous un nouveau nom (“Sauvegarder sous” dans le menu), par exemple `bonjourmonde.tex`.
2. Ajoutez le support linguistique en important le paquet (*package*) adéquat et en lui passant l’option appropriée pour un document en français (*c.f.* `cours-td`).
3. Pour vérifier que le langage français a bien été chargé, utilisez la commande `\language` définie par le paquet `babel`, qui affiche (dans le contenu) le nom du langage courant. Quel est le nom de ce langage ?
4. Si vous observez le *log*<sup>4</sup> de la compilation précédente, vous devriez remarquer un *Warning* concernant le choix inapproprié de police (*font*) pour la sortie `pdf` en français. Comme suggéré par le *Warning*<sup>5</sup>, importez le paquet `fontenc` (encodage de police, pour la sortie `pdf`) avec l’option `T1`.
5. Changez le texte “*Hello World!*” par “*Bonjour monde !*”. Compilez et regardez.
6. Changez le texte en “*Bonjour monde étoilé !*”. Compilez ; visualisez.

---

*Les accents (et autres caractères particuliers) ne sont pas supportés par défaut dans le code source L<sup>A</sup>T<sub>E</sub>X, lorsqu’on utilise de vieilles versions de L<sup>A</sup>T<sub>E</sub>X. Ils peuvent néanmoins être réalisés par des commandes (e.g., `\`e` donne é, `\`e` donne ê), ou – plus simple et préférable – écrits directement grâce au paquet `inputenc` (encodage d’entrée) que l’on importe avec l’option `utf8` – c.f. ci-dessous. Si cet encodage est supporté nativement par L<sup>A</sup>T<sub>E</sub>X depuis 2018 – et en particulier par les versions installées au SCI et à l’ISIMA – importer le paquet avec l’option adéquat reste une bonne pratique, pour des raisons de portabilité, et de rétro- & post-compatibilité.*

---

7. Ajoutez `\usepackage[utf8]{inputenc}` au préambule et recompilez.
8. Ajoutez du contenu dans votre article (remplissez au moins deux pages), par exemple à l’aide du paquet `blindtext` et de la commande `\blindtext`<sup>6</sup> qu’il définit, et qui permet de générer des *paragraphes de texte*.

---

3. Voyez la Section “Sauvegarder vos données” du document d’instruction, afin d’éviter une perte de données d’un TP à l’autre, d’une machine à l’autre.

4. Les *log* de L<sup>A</sup>T<sub>E</sub>X sont en général assez peu lisibles.

5. Afin de voir l’intégralité des *log*, compilez votre fichier, à l’aide de la commande `pdflatex` depuis le terminal, et/ou, observez le fichier d’extension `.log` créé à côté du fichier source par la compilation.

6. La commande `\blindtext` accepte un argument optionnel, donné entre crochets, qui doit être un nombre indiquant combien de paragraphes doivent être générés.

9. Changez la classe du document, pour utiliser la classe nommée `article`<sup>7</sup>. Compilez. Voyez-vous un changement ?
10. Définissez un titre, un auteur et une date dans le préambule (*c.f.* exemple final du cours-td). Compilez. Voyez-vous des changements ?
11. Faites afficher automatiquement le titre à partir des méta-données spécifiées à la question précédente, à l'aide de la commande vue en cours-td. (Cette commande n'est pas une commande de base de L<sup>A</sup>T<sub>E</sub>X, mais est définie dans la classe `article`.)
12. Les classes sont comme des paquets, elles admettent des options (données entre crochet, séparées par des virgules). Testez indépendamment les deux options `titlepage` et `twocolumn` acceptées par la classe `article`, et expliquez, dans le fichier `tex` ce qu'elles font.
13. Ajoutez l'option `thisisnotanoption` qui de toute évidence n'est pas connue de la classe `article` et compilez. Que se passe-t-il ?

---

*Les classes acceptent n'importe quelle option, les options inconnues sont juste ignorées par la classe, mais peuvent être utilisées par les paquets. Ceci permet de donner une seule fois une option qui pourrait intéresser plusieurs paquets.*

---

14. Retirez l'option `french` de l'import du paquet `babel` que vous devriez avoir importé à la [question 2](#), et ajoutez la à la classe. Compilez et vérifiez<sup>8</sup> que le langage chargé est bien le français.

## Exercice 3 *Oh la belle réf!*

### PRÉLIMINAIRES

1. Ouvrez un nouveau fichier, et créez un préambule utilisant la classe `article` et un document vide. Votre document doit pouvoir être compiler –après avoir été sauvegardé sous un nom approprié– mais n'ayant pas de contenu, il ne devrait pas produire de fichier `pdf`.<sup>9</sup>
2. Ajoutez un titre, un auteur et une date, et faites en sorte que l'entête de titre s'affiche sur la première page.

### STRUCTURATION DU DOCUMENT

Nous allons maintenant voir comment créer des parties dans notre document.

3. Utilisez la commande `\section` (qui prend en argument, entre accolades, le titre de la section) pour créer trois sections (avec des titres de votre choix).

---

7. Ça, c'est une vraie classe – *c.f.* <sup>1</sup> – qu'on utilisera beaucoup !

8. Utilisez pour cela la commande vue à la [question 3](#).

9. `texmaker` peut se plaindre parce qu'il ne trouve pas de fichier `pdf` à ouvrir : c'est l'ouverture du `pdf` qui échoue, pas la compilation.

4. Utilisez la commande `\subsection` (qui fonctionne de manière similaire) pour créer deux sous-sections, dans chacune des trois sections.
5. Créez deux sous-sous-sections dans la première sous-section (devinez comment).
6. Placez un peu de texte<sup>10</sup> dans les sections de façon à ce que le titre de la deuxième section apparaisse sur une seconde page.

## TABLE DES MATIÈRES

7. À l'aide de la commande `\tableofcontents`, placez la table des matières juste après l'entête de titre. Compilez (éventuellement deux fois). Que voyez-vous ?
8. Remarquez l'impact de l'import de `babel` (avec l'option `french`) sur le résultat de la question précédente. (Compilez, plusieurs fois, avec et sans cet import et observez la différence.)
9. Pour indiquer à L<sup>A</sup>T<sub>E</sub>X jusqu'à quel niveau de section il doit aller dans la table des matières, il faut modifier la valeur d'un compteur interne nommé `tocdepth` (*toc* sont les initiales de *table of contents* et *depth* signifie profondeur en anglais). Ajoutez la commande suivante dans le préambule : `\setcounter{tocdepth}{2}`. Compilez, observez. Que doit-on changer pour ne voir que les sections (et pas les sous-sections) ?
10. La commande `\section` et ses variantes (`\subsection`, `\subsubsection`, ...) acceptent un second argument, qui est optionnel. Un tel argument est donné entre crochets avant l'argument obligatoire qui est donné entre accolades (*e.g.*, `\section[argument optionnel]{argument obligatoire}`) – c'est une forme standard que vous avez déjà rencontrée et utilisée avec les commandes `\documentclass` et `\usepackage`. Essayez de donner une valeur à l'argument optionnel de l'une de vos commandes `\section`, compilez (éventuellement deux fois), visualisez, et comprenez le rôle de cet argument optionnel.

## RÉFÉRENCES

Nous allons maintenant faire référence à la dernière sous-section (de la dernière section). Ceci fait appel au mécanisme de référencement interne de L<sup>A</sup>T<sub>E</sub>X, qui fonctionne de manière identique quelque soit le type de référence (une référence à une section, à un exemple, à un élément d'une liste numérotée, à une figure, *etc.* ...). Pour cela, il faut d'une part définir une étiquette ([question 11](#)), et d'autre part y faire référence ([question 12](#)).

11. Définissez une étiquette dans l'élément que l'on veut référencer (ici, la dernière sous-section) avec la commande `\label{sec/etiq}`. Le mot `sec/etiq` doit être choisi, de manière à être unique dans le document : c'est une *clef d'identification*.<sup>11</sup>

---

10. Vous pouvez utiliser le paquet `blindtext` vu à la [question 8](#) de l'[exercice 2](#).

11. C'est une bonne pratique que de préfixer cette étiquette par `sec/` pour indiquer que l'élément référençable est de type section ; on peut également préciser qu'il s'agit d'une sous-section avec le préfixe `ssec/`. Ici, le caractère `/` délimite la fin de ce préfixe ; souvent, le caractère `:` est utilisé à cet effet, mais il est déconseillé pour des documents en français, car il est "modifié" par `babel` pour être traité correctement (en typographie française, `:` doit être précédé d'une espace insécable fine, et `babel` s'assure de cela).

12. Faites référence (par exemple dans la première section) à cette étiquette, *via* la commande `\ref{sec/etiq}` (le mot `sec/etiq` doit être le même que celui choisi pour l'argument de la commande `\label` à la question précédente). Vous pouvez par exemple écrire, dans la première section :

Nous verrons bien dans la Section `\ref{sec/etiq}`.

Compilez (éventuellement deux fois) ; visualisez.

13. Il est bien entendu possible de faire plusieurs fois référence au même élément (avec la même étiquette). Faites-le, dans une autre section.
14. Ajoutez une section, avant la dernière section, afin de changer le numéro de la section référencée. Compilez (éventuellement deux fois) ; observez ; applaudissez.
15. Naturellement, il est possible de référencer d'autres éléments : il suffit pour cela de placer d'autres étiquettes (`\label`) avec des clefs d'identification différentes. Référez deux autres sections (ou sous-sections).
16. Que se passe-t-il si la clef donnée à (`\ref`) n'a pas été définie dans le document ?

## HYPERLIENS

17. Importez le paquet `hyperref` (dans le préambule). Compilez. Cliquez, dans le pdf, sur le numéro de la référence introduite à la **question 12**. Que se passe-t-il ?

Le paquet `hyperref` gère en effet les liens hypertexte (*i.e.*, cliquables), qu'ils soient interne (*i.e.*, ils pointent vers un emplacement du document, comme les références des questions précédentes) ou externe (*i.e.*, ils pointent vers une *url* externe).

18. Créez un lien vers **la page wikipedia** qui vous explique l'origine du *Lorem ipsum*, à l'aide de la commande `\url` (1 argument : l'*url* pointée), puis à l'aide de la commande `\href` (2 arguments : l'*url* pointée et le texte à afficher).

## Exercice 4      *Des maths, des maths !*

1. Créez un nouveau fichier, et sauvegardez le sous un nom adéquat, par exemple `math.tex`. Écrivez-y un préambule fonctionnel.
2. Écrivez le texte suivant : “ $3-2+4=5$ ”. Compilez ; visualisez.

En  $\text{\LaTeX}$ , les formules mathématiques ont une place de choix ( $\text{\LaTeX}$  a été créé en grande partie pour permettre d'écrire de jolies formules mathématiques facilement). Lorsqu'on écrit ces formules, on utilise le *mode math*. Pour passer en mode math (un mode où le rendu est orienté math, et où des commandes spécifiques aux maths sont disponibles) il suffit d'utiliser le caractère spécial `$`. Pour en sortir, le même caractère fait l'affaire (à condition qu'il ne soit pas collé au précédent `$`).

3. Écrivez, sous le texte de la question précédente, la même formule, mais cette fois-ci en mode math. Compilez ; observez ; comparez.

4. Essayez le code suivant, en mode normal puis en mode math : “ $2^{10}=1024$ ”.<sup>12</sup>  
Que se passe-t-il (en mode normal) ? À quoi sert l’accent circonflexe (^) ?
5. En math, il est fréquent d’utiliser des variables indicées (*e.g.*,  $a_1, a_2, \dots, a_n$ ). Les indices s’obtiennent de la même manière que les exposants, mais avec le symbole `_`. Écrivez :  $a_1, a_2, \dots, a_n$ , sachant que ‘ $\dots$ ’ s’obtient avec la commande `\ldots`.
6. À l’aide des commandes `\times` et `\cdots`, écrivez le produit des variables de la question précédente :  $a_1 \times a_2 \times \cdots \times a_n$ .
7. Écrivez des inégalités strictes (*e.g.*,  $3 < 4$  et  $5 > 4$ ). Que font les commandes `\leq` et `\geq` (disponibles uniquement en mode math) ?
8. D’autres symboles mathématiques existent, comme ceux ci-dessous. Essayez-les.
  - les quantificateurs : `\exists` et `\forall`;
  - les relations `\neq`, `\equiv`, `\iff`<sup>13</sup> `\sim`, et `\hookrightarrow`;
  - les fonctions et opérateurs `\cos`, `\sin`, `\tan` et `\lim`;
  - le constructeur de fraction `\frac` qui prend deux arguments et que nous avons vu en cours-td;
  - des constantes, comme `\infty`.
9. Des symboles plus avancés sont disponibles, avec les paquets `mathtools` et `amssymb`. Importez ces derniers, et testez les commandes `\nleq`, `\ngeq` et `\implies`.  
(**Bonus** : Trouvez et indiquez pour chacune d’elle le paquet qui la définit.)
10. Parfois, pour augmenter la lisibilité, nous voulons écrire des équations sur des lignes à part. Ceci est possible grâce à l’environnement `equation*`, défini par le paquet `mathtools`, qui passe automatiquement son contenu en mode math. Produisez l’équation suivante :
 
$$e = mc^2$$
11. L’astérisque à la fin du nom `equation*` permet en fait d’utiliser une variation de l’environnement `equation`. Testez cet autre environnement (donnez ainsi plusieurs équations, chacune dans son environnement propre), et expliquez la différence entre `equation` et `equation*`.
12. Définissez une étiquette (avec la commande `\label`) à l’intérieur d’une equation (donnée avec l’environnement `equation` sans astérisque) et faites-y référence dans du texte.<sup>14</sup> Compilez (éventuellement deux fois) ; observez.
13. Modifiez la référence créée à la question précédente afin d’utiliser la commande `\eqref` plutôt que `\ref`. Compilez. Quel changement observez-vous ?
14. Ajoutez une équation avant celle référencée, puis compilez et observez.
15. En mode math, le texte est formaté différemment et les espaces sont supprimés. Il est cependant possible – et souvent utile – d’écrire du texte, en utilisant la

---

12. Pour obtenir l’accent circonflexe seul, il faut presser deux fois la touche de clavier correspondante.

13. *iff* est l’abréviation de *if and only if*, ce qui signifie si et seulement si.

14. *C.f.* [exercice 3](#).

commande `\text` qui prend en argument le texte à écrire. Écrivez la formule suivante (dans l'environnement adéquat) :

$$\forall k < n, k \text{ est un diviseur de } n!$$

16. Dans une nouvelle section, écrivez les énoncés et les solutions des questions sur les quantificateurs vues en cours-td.
17. En utilisant la notation des indices, et, entre autres, les commandes (mode math) `\to`, `\theta` et `\pi`, écrivez l'équation suivante :

$$\lim_{\theta \rightarrow (\frac{\pi}{2})^-} \tan(\theta) = \infty$$