Karnaj

Zeste de savoir

Écrire des lettres en LaTeX

29 décembre 2018

Table des matières

1. Introduction

Vous connaissez LaTeX et vous voulez apprendre à écrire des lettres en l'utilisant? Vous êtes au bon endroit. Du fait de sa nature plus sémantique que syntaxique, LaTeX est particulièrement adapté à l'écriture de lettres. Imaginez un document avec un code de ce genre.

```
\adresse{Chez moi}
\nom{Karnaj}
\destinataire{Zeste de Savoir}
```

Avec des commandes qui se chargent de placer les éléments aux bons endroits, on peut alors se concentrer sur le contenu de la lettre, et ne pas se poser de questions. Les éléments sont bien placés avec le bon espacement, il n'y a plus besoin de se demander où placer le destinataire. À gauche, à droite, avant la date ou après?

La plupart des logiciels WYSIWYG posent ce genre de problèmes et vous obligent à prendre le temps de placer à peu près bien les différents éléments. En ce sens, écrire vos lettres avec LaTeX peut s'avérer être un gain de temps, et vous permet d'obtenir une bonne base commune pour vos lettres.

Dans ce tutoriel, nous allons donc voir comment écrire une lettre avec LaTeX.

Prérequis
Connaissance des bases du LaTeX.
Objectifs
Découvrir un moyen simple d'écrire des lettres en LaTeX.
Présenter une méthode pour faire du publipostage (envoyer une lettre à plusieurs personnes).
Aspects non traités
Personnalisation de la lettre.
Ajout d'une icône à la lettre.

2. La classe lettre

Dans ce tutoriel, nous allons nous intéresser à la classe `lettre`, une classe spécialement créée pour écrire des lettres.

2.1. La classe `letter` et la classe `lettre`

La classe `letter` est aussi une classe faite pour écrire des lettres. Elle offre un certain nombre de commandes (de la même manière que le fait le code en introduction) pour formater nos lettres. Elle permet donc de placer par exemple:

— le nom de l'expéditeur;
— l'adresse de l'expéditeur;
— la signature de l'expéditeur;
— le nom du destinataire;
— l'adresse du destinataire;
— la date;
— le lieu;
— la formule de politesse.

Et ce n'est qu'une petite partie de ce qu'elle permet de placer. C'est une classe vraiment complète.

Oui d'accord, mais pourquoi parlons-nous de la classe `letter`. Ne devions-nous pas voir la classe `lettre`?

Bonne question. Mais la question pertinente est surtout: **si la classe `letter` permet déjà de faire tout ce que l'on veut, et qu'en plus elle est complète, pourquoi allons-nous utiliser une autre classe?**

En fait, la classe `letter` est très complète, mais elle ne respecte pas les conventions françaises, mais les conventions anglophones. La classe `lettre` a été développée pour répondre aux mieux

à nos conventions. Elle s'appuie sur la classe `letter` et la complète. Certaines commandes sont donc communes aux deux classes. Si vous avez déjà de l'expérience avec la classe `letter`, vous ne devriez pas être trop dépaysé.

La classe `lettre` prend aussi en charge les conventions allemandes, américaines et anglaises.

Dans ce tutoriel, nous n'allons pas aborder de façon complète cette classe, nous n'allons que présenter les options les plus communes.

2.2. La classe `lettre`

Maintenant que nous avons mis au clair la raison pour laquelle nous allons utiliser la classe `lettre`, voyons le début de son utilisation. Le code minimal est celui-ci.

```
1 \documentclass[12pt]{lettre}
2
3 \begin{document}
4
5 \end{document}
```

La classe `lettre` a les options habituelles:

- `10pt`, `11pt` et `12pt` pour la taille du texte avec `10pt` la taille par défaut;
- `a4paper` pour les dimensions de la feuille;
- `oneside` et `twoside` pour obtenir un document en recto ou en recto-verso (une lettre est la plupart du temps en recto), l'option par défaut étant `oneside`.

Vous disposez également des options `draft` et `final`.

Finalement, pour l'instant, nous n'avons rien vu de nouveau. La classe `lettre` se comporte comme une banale classe `article` pour le moment.

Pour accéder à toutes les bonnes fonctions, le code minimal pour LaTeX estle suivant.

```
1 \documentclass[12pt]{lettre}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4
5 \begin{document}
6
7 \end{document}
```

Et pour XeLaTeX…

```latex
\documentclass[12pt]{lettre}
\usepackage{fontspec}

\begin{document}

\end{document}
```

Nous avons maintenant notre préambule minimal. Nous conseillons de le compléter avec le *package* **babel** (avec LaTeX) ou **polyglossia** (avec XeLaTeX ou LuaLaTeX).

À partir de maintenant, nous ne mettrons plus le préambule mais seulement le code entre le `\begin{document}` et le `\end{document}`.

2.2.1. L'environnement `letter`

Le cœur d'une lettre avec la classe `lettre` est l'environnement `letter`. Il permet… d'écrire une lettre. Dans un document, nous pouvons avoir plusieurs environnements `letter`. Ce seront autant de lettres que nous obtiendrons. Nos documents ressembleront donc à ceci.

```latex
\begin{letter}
% Contenu de notre lettre.
\end{letter}

\begin{letter}
% Contenu d'une autre lettre.
\end{letter}
```

Dans la prochaine partie, nous verrons comment utiliser cet environnement et créer nos lettres.

3. Écrire nos lettres

3.1. Les informations nécessaires

L'environnement `letter` nous permet d'écrire un document très sémantique. Tout d'abord, il prend comme paramètres le nom et l'adresse du destinataire de la lettre. On les sépare de deux antislashs `\\` (qui permettent le retour à la ligne). Dans l'adresse, on va également à la ligne, généralement.

```
\begin{letter}{nom du destinataire\\
               adresse\\
               du\\
               destinataire
              }
\end{letter}
```

Ces informations sont placées en haut à droite de la lettre.

Vous pouvez ensuite — juste après, dans l'environnement `letter` — indiquer la langue de votre lettre avec l'une de ces commandes:

- `\francais` (par défaut);
- `\allemand`;
- `\anglais`;
- `\americain`;
- `\romand`.

La classe se chargera de placer les différents éléments en fonction de la langue que vous avez indiquée et d'écrire par exemple «Subject» ou «Objet».

Dans ce tutoriel, vous aurez le placement de plusieurs données. Il s'agit du placement français.

Vous disposez de deux commandes pour préciser la date et le lieu d'envoi de la lettre. Ces deux données sont placées en haut à droite de la lettre, avant l'adresse du destinataire. Elles sont affichées de cette manière: «lieu, date».

La commande `\lieu` est facultative. Si elle n'est pas utilisée, un lieu par défaut sera utilisé. Pour ne pas obtenir de lieu, il faut utiliser la commande `\nolieu`.

La commande `\date` est elle aussi facultative. Si elle n'est pas utilisée, c'est la date de compilation — dans la langue choisie — qui sera affichée. Pour ne pas afficher de date, il faut utiliser la commande `\nodate`.

Vous pouvez faire en sorte de conserver la même date pour une lettre, même lors de compilations ultérieures. Pour cela, vous devez passer `origdate` en paramètre facultatif du `\documentclass`.

Dans ce cas, un fichier `.odt` sera créé. Il contiendra la date. Si lors de la compilation le fichier existe, la date sera récupérée, sinon, une nouvelle date sera créée (en fonction du paramètre passé à la commande `\date`) et le fichier sera créé.

3.2. Les données sur l'expéditeur

Après être entré dans l'environnement, vous disposez de plusieurs commandes pour donner des informations à propos de l'expéditeur.

— La commande `\address` prend en paramètre l'adresse de l'expéditeur (elle sera placée en haut à gauche de la lettre). Vous pouvez la formater comme vous voulez, la classe se charge juste de la placer en haut à gauche. De la même manière que pour l'adresse du destinataire, il faut utiliser le double antislash pour aller à la ligne. Si vous ne fournissez pas d'adresse, une adresse par défaut sera affichée (l'adresse par défaut — et en fait tous les paramètres par défaut — sont placés dans un fichier `ins` placé dans le répertoire de la classe `lettre`).
— Le numéro de téléphone de l'expéditeur est donné en paramètre de la commande `\telephone`. Il est placé juste en dessous de son adresse. Si vous ne précisez pas de numéro, un numéro par défaut sera affiché. Si vous ne voulez pas avoir de téléphone sur la lettre, vous devez utiliser la commande `\notelephone`.
— La commande `\fax` fonctionne comme la commande `\telephone`. Son paramètre est le numéro de fax de l'expéditeur. S'il n'y en a pas, celui par défaut sera affiché, et si vous n'en voulez pas, il faut utiliser la commande `\nofax`.
— La commande `\email` prend en paramètre l'adresse courriel de l'expéditeur. Elle l'affiche en dessous des numéros de téléphone et de fax s'ils sont disponibles. Sinon, elle est en dessous de l'adresse. L'adresse électronique est facultative, si vous n'en fournissez pas, il n'y en aura pas d'affichée.

> *i* Notez qu'une ligne vide est laissée entre l'adresse et ce qui la suit (numéros et courriel).

Ces commandes vous permettent de définir complètement l'expéditeur de la lettre. Il ne manque qu'une chose: son nom.

Le nom de l'expéditeur est donné en paramètre de la commande `\name`. Pour indiquer sa signature, il faut utiliser la commande `\signature` qui prend sa signature en paramètre. Si elle n'est pas utilisée, la signature par défaut est le nom indiqué par `\name` (pour ne pas avoir de signature, il faut donner à la commande un argument vide). Elle s'affiche en dessous de la lettre à droite, à la place de la signature.

3.3. La lettre

Maintenant que nous avons bien vu tout ce qu'il faut à propos des informations sur le destinataire et l'expéditeur, il ne nous reste plus qu'à écrire le corps de notre lettre.

Si vous avez déjà essayé de compiler un code avec les commandes précédentes, vous avez remarqué que rien ne s'affichait. Aucune adresse, aucune date, aucun numéro… C'est tout à fait normal. Pour vraiment commencer la lettre, il faut utiliser la commande `\opening`. C'est elle qui se charge de mettre en forme toutes les données précédemment transmises. Cette commande prend en paramètre les salutations que vous voulez placer en début de lettre. Par exemple…

```
\opening{Madame, Monsieur, }
```

Après cette commande, vous pouvez commencer à écrire le corps de votre lettre. Vous pouvez écrire tout ce que vous voulez et le mettre en forme comme vous voulez. Les environnements fonctionnent, les mathématiques fonctionnent, c'est votre espace de liberté.

Après avoir écrit tout ce que vous aviez à écrire, il faut ensuite terminer votre lettre et saluer votre interlocuteur. La commande `\closing` se charge de cela, Elle prend en paramètre votre formule de politesse, qu'elle met en forme. Elle se charge également de placer votre signature en dessous de votre formule de politesse.

Finalement, voici un squelette de lettre.

```
\begin{letter}{nom destinataire\\adresse destinataire}
   \date{date envoi}                          % \nodate pour pas de
       date.
   \lieu{lieu envoi}                          % \nolieu pour pas de
       lieu.
   \name{nom expéditeur}
   \address{adresse expéditeur}
   \telephone{numéro téléphone expéditeur} % \notelephone pour pas
       de téléphone.
   \fax{numéro fax expéditeur}                % \nofax pour pas de
       fax.
   \email{courriel expéditeur}                % Ne rien mettre si on
       ne veut pas de courriel.
   \opening{salutations, }
      On écrit le contenu de la lettre ici.
   \closing{formule de politesse}
\end{letter}
```

Vous pouvez compiler ce code et observer le résultat que vous obtenez.

Essayez-donc de choisir une autre langue, disons l'anglais. Il suffit de rajouter la commande `\anglais` juste après le `\begin{letter}`. Après compilation, vous pouvez observer les différences entre le document français et le document anglais.

3.4. Suppléments

Nous avons maintenant un code complet. Alors rendons-le encore plus complet. Pour cela, nous allons utiliser d'autres commandes (oui, encore) qui nous permettrons de rajouter des précisions.

L'une des commandes supplémentaires les plus utiles est la commande `\conc`. Elle est à placer juste avant la commande `\opening`, et prend en paramètre le sujet de votre lettre, qu'elle met en forme et affiche (l'affichage est de la forme «Objet: sujet de la lettre» en français).

De la même manière, la commande `\ps` permet d'écrire un texte avec un label. Cette commande prend deux paramètres: le label à afficher et le texte à afficher. Elle s'utilise donc avec ce code: `\ps{label}{texte}`. Elle est par exemple utilisée pour écrire des post-scriptum. Cette commande peut être utilisée entre les salutations (la commande `\opening`) et la formule de politesse (la commande `closing`) ; auquel cas, le label et le texte seront affichés dans la lettre, mais elle est plus usuellement employée après la formule de politesse, comme il est d'usage pour les post-scriptum.

4. Exemple de lettre

À titre d'exemple, nous allons écrire une lettre à Clem, notre très chère mascotte qui jette tous les jours sur nous un zeste de fraîcheur et de connaissance.

4.1. Consignes

Vous devez juste reproduire cette lettre.

Caverne des Dieux
29 Cascade des Bonheurs

Lieu sombre et inconnu, le 07 septembre 2014

À Clem
Mascotte officielle du ZdS
Plage de Développement

Très chère Clem,

Nous venons d'apprendre que des développeurs sont venus te chercher hier, pour t'emmener à la Plage de Développement en vue de devenir la mascotte du bien-aimé site Zeste de Savoir. Félicitations pour ton nouveau titre. Nous sommes sûrs que tu t'amuses beaucoup avec eux. Les caribous sont dignes de confiance, tu peux leur confier tes inquiétudes quant à l'évolution du site. Même si les renards et les loups semblent moins avenants, tu peux aussi leur parler, ils ne mordent pas (ou alors très rarement). En plus, il paraît même qu'il y en a qui sont des dieux galactiques !

Voilà, nous espérons tout de même que tu n'es pas trop dépaysée. N'oublie pas de manger tes cinq fruits et légumes par jour et grandis bien.

Qu'un Zeste de fraîcheur et de connaissance t'accompagne durant ton aventure.

Karnaj et autres

PS : Fais attention, il est venu à notre attention qu'un certain saucisson pourrait ne pas très bien prendre ta nomination. Nous espérons que cette lettre arrivera à temps pour te prévenir.

FIGURE 4. – Lettre à Clem.

4.2. Corrigé

Bon, voici un corrigé, mais normalement, nous devons déjà avoir un code fonctionnel.

```
\documentclass[12pt]{lettre}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[french]{babel}

\begin{document}
   \begin{letter}{À Clem\\
                  Mascotte officielle du ZdS\\
                  Plage de Développement\\
                  }
      \date{le 07 septembre 2014}
      \lieu{Lieu sombre et inconnu}
      \name{Karnaj et autres}
      \address{Caverne des Dieux\\
               29 Cascade des Bonheurs}
      \notelephone
      \nofax
      \opening{Très chère Clem, }
         Nous venons d'apprendre que des développeurs sont venus te
             chercher hier, pour
         t'emmener à la Plage de Développement en vue de devenir la
             mascotte du bien-aimé
         site Zeste de Savoir. Félicitations pour ton nouveau titre.
             Nous sommes sûrs
         que tu t'amuses beaucoup avec eux. Les caribous sont
             dignes de confiance, tu
         peux leur confier tes inquiétudes quant à l'évolution du
             site. Même si les
         renards et les loups semblent moins avenants, tu peux
             aussi leur parler, ils
         ne mordent pas (ou alors très rarement). En plus, il
             paraît même qu'il y en a
         qui sont des dieux galactiques !

         Voilà, nous espérons tout de même que tu n'es pas trop
             dépaysée. N'oublie
         pas de manger tes cinq fruits et légumes par jour et
             grandis bien.
      \closing{Qu'un Zeste de fraîcheur et de connaissance
          t'accompagne durant ton aventure.}
      \ps{PS :}{Fais attention, il est venu à notre attention qu'un
          certain saucisson pourrait
          ne pas très bien prendre ta nomination. Nous espérons que
              cette lettre arrivera
          à temps pour te prévenir.}
```

```
34      \end{letter}
35  \end{document}
```

5. Pour aller plus loin

5.1. Le fichier `ins`

Lorsque vous écrivez des lettres, votre nom reste sûrement le même. Et à moins que vous ne déménagiez, votre adresse aussi. En fait, plein de détails restent les mêmes. Alors on aimerait bien ne pas avoir à les réécrire tout le temps. Ce serait un gain de temps appréciable. On sait que `\address` et `\name`, par exemple, ont des valeurs par défaut. Il suffit de changer ces valeurs par défaut et le tour est joué.

Vous n'avez pas vraiment pensé à faire ça? On peut modifier directement les fichiers des classes, mais il vaut mieux les laisser comme ils sont.

Plutôt que de modifier les valeurs par défaut, nous allons profiter d'une des options de la classe `lettre`. Nous allons créer un fichier à l'extension `.ins`. Dans ce fichier, nous pourrons écrire les quelques lignes fixes de notre code. Ce fichier est ensuite appelé à l'aide de la commande `\institut`.

Ce fichier doit être placé dans le même dossier que le fichier dans lequel il est appelé.

Voici à quoi pourrait ressemblerait un fichier `.ins`.

```
1  \name{expéditeur}
2  \address{expéditeur}
3  \telephone{numéro de téléphone}
4  \email{adresse courriel}
5  \nofax
```

On écrit exactement ce que l'on aurait écrit dans le fichier `.tex`.

Dans le fichier `.tex`, il suffit de supprimer les lignes que nous avons placées dans le fichier `.ins` et d'utiliser la commande `\institut{nom du fichier .ins}` avant l'environnement `letter`.

Il ne faut pas mettre l'extension. Si votre fichier s'appelle `lettre.ins`, la commande à utiliser est `\institut{lettre}`.

Une fois que ce fichier a été écrit, il ne vous reste plus qu'à l'utiliser pour toutes vos lettres.

De plus, si une commande du fichier `.ins` apparaît dans le fichier principal, c'est la commande du fichier principal qui sera prise en compte, nul besoin de modifier le fichier `ins`. Donc, si un jour vous voulez écrire une lettre en utilisant, par exemple, une adresse électronique différente (supposons que vous disposiez d'une adresse électronique personnelle et d'une autre professionnelle), vous n'aurez qu'à rajouter la commande `\email` dans votre fichier `.tex`.

5.2. Le trait de pliage

Vous n'avez pas pu manquer ce petit trait noir que la classe `lettre` place sur le document. Il s'agit du trait de pliage. Il permet de faciliter le pliage de la lettre en trois pour une mise en enveloppe. Ce trait noir indique l'endroit où se fait le premier pli de manière à ce que l'adresse, la date et le lieu tombent exactement en face de l'ouverture de la lettre.

Vu comment c'est compliqué à faire soi-même, ce trait est d'une grande aide. Mais il est parfois inutile, et peut-être voudriez-vous l'enlever. Pour cela, nous allons définir une commande (généralement on l'appelle `\NoRule`). La voici.

```
\makeatletter
   \newcommand*{\NoRule}{\renewcommand*{\rule@length}{0}}
\makeatother
```

Il suffit ensuite d'utiliser la commande `\NoRule` juste après être entré dans l'environnement `letter`.

Vous pouvez aussi placer la ligne `\renewcommand*{\rule@length}{0}` dans votre fichier `.ins`. Ainsi, dès que vous utiliserez ce fichier `.ins`, vous n'aurez pas de trait de pliage.

5.3. Le publipostage

Lorsqu'une lettre doit être envoyée à plusieurs personnes, il y a quelques mots seulement qui changent. Le nom et le prénom de la personne, son adresse, l'apostrophe (les «Cher Monsieur» ou «Mon très cher»). S'il n'y a qu'une dizaine de personnes, vous pouvez changer ces données à la main et recompiler à chaque fois, mais cela prend déjà du temps. Alors imaginez le temps pris pour une centaine de lettres voire plus !

Une autre solution serait d'écrire un script en bash, par exemple, qui se chargerait de remplacer les données qui changent et de faire la nouvelle compilation.

En fait, il existe même des *packages* en LaTeX qui se chargent de cela pour nous. C'est cette solution que nous allons voir en utilisant le *package* **datatool**. Ce *package* nous permet de faire des boucles et de récupérer des données dans un fichier. Ce sont ces fonctionnalités que nous allons utiliser.

Pour plus d'informations sur le *package* datatool, allez voir la documentation .

Voici comment nous allons fonctionner.

1. Nous allons créer un fichier CSV (avec un éditeur de texte quelconque, ou LibreOffice, par exemple). Dans celui-ci, nous allons placer nos données de cette manière: sur la première ligne, les différentes étiquettes (« nom », « prénom », « adresse », etc.) et sur les lignes suivantes les noms, prénoms, etc. des différentes personnes qui doivent recevoir la lettre.
2. Parcourir les différentes données du fichier avec datatool et créer une lettre par ligne.

Voyons un exemple de code, puis analysons-le.

```
\begin{document}
   \DTLloaddb{fichier}{fichier.csv} % On charge le fichier .csv.
   \DTLforeach{fichier}{\apostrophe =Apostrophe, \prenom =Prénom,
       \nom =Nom, \adresse =Adresse}
   {                                % On parcourt tout le fichier.
      \begin{letter}{À \prenom \bsc{\nom}\\
                      \adresse
                     }
         \date{le 07 septembre 2014}
         \lieu{Lieu sombre et inconnu}
         \name{Karnaj et autres}
         \address{Caverne des Dieux\\
                  29 Cascade des Bonheurs}
         \notelephone
         \nofax
         \opening{\apostrophe \prenom, }
            Nous venons d'apprendre que des développeurs sont venus
                te chercher hier, pour
            t'emmener à la Plage de Développement en vue de devenir
                la mascotte du bien-aimé
            site Zeste de Savoir. Félicitations pour ton nouveau
                titre.
         \closing{Qu'un Zeste de fraîcheur et de connaissance
             t'accompagne durant ton aventure.}
      \end{letter}
   }
\end{document}
```

Ce code n'est pas trop compliqué. Avec la commande `\DTLloaddb` on charge le fichier `fichier.csv`. Le premier argument de la commande est le nom qu'on veut donner aux données qu'on utilisera et son second argument est le nom du fichier à charger.

Ensuite, la commande `\DTLforeach` nous permet de parcourir tous les éléments du premier argument (dans notre code, le premier argument est `fichier`, c'est-à-dire les données chargées). Le second argument permet de définir des commandes:

— `apostrophe` correspondra à la donnée «Apostrophe» qu'on parcourt;
— `prenom` correspondra à la donnée «Prénom» qu'on parcourt;
— `nom` correspondra à la donnée «Nom» qu'on parcourt;
— `adresse` correspondra à la donnée «Adresse» qu'on parcourt.

Ainsi, à chaque tour de boucle, ces commandes prendront une valeur différente qui correspondra aux données d'un nouveau destinataire.

N'oubliez pas d'ajouter `\usepackage{datatool}` à votre code.

Ce code produira alors autant de lettres qu'il le faut. Un fichier `csv` qui pourrait être associé au code précédent estcelui-ci.

Apostrophe	Prénom	Nom	Adresse
Très chère	**Clem**		**ZdS**
Mon bon ami	Pierre	Pierre	Bonne question

Testez donc le code avec ces données. Vous obtiendrez deux lettres (dans un seul fichier PDF).

Le *package* datatool permet même de faire des structures conditionnelles. Par exemple, supposons que dans votre fichier CSV, vous ayez une zone « Sexe » dans laquelle vous ayez écrit «F» si le destinataire est une femme et «G» sinon. Vous aimeriez bien avoir «À monsieur» ou «À madame» en fonction de ce paramètre, non?

Pour cela, il vous suffit d'utiliser la commande `\DTLifstringeq` qui permet de tester l'égalité de deux chaînes, de faire quelque chose si elles sont égales et quelque chose d'autre sinon. Le code qu'on obtient estcelui-ci.

```
\begin{letter}{À \DTLifstringeq*{\sexe}{G}{Monsieur}{Madame}
    \prenom \bsc{\nom}\\
                \adresse
              }
```

Les deux premiers paramètres de la commande sont les deux chaînes à comparer. Le troisième paramètre est ce qu'il faut faire si les chaînes sont égales et le quatrième ce qu'il faut faire si elles ne le sont pas. Ici, on met «Monsieur» si `\sexe` vaut «G», sinon, on met «Madame».

La version étoilée de `\DTLifstringeq` permet de ne pas tenir compte de la casse et donc d'écrire «G», «g», «f» ou «F».

Notez que le *package* datatool permet de faire d'autres types de comparaisons. Nous ne pouvons que renvoyer encore une fois à la documentation pour voir toutes ses fonctionnalités.

6. Conclusion

Vous avez maintenant de quoi écrire de belles lettres. Mais l'aventure n'est pas finie. Je vous renvoie à la documentation de la classe `lettre` pour plus d'informations à son sujet. Elle est en français, en plus. Grâce à elle, vous verrez comment ajouter une image (l'icône officielle de l'institut qui envoie une lettre, par exemple) et comment personnaliser votre lettre.

Vous pouvez aussi vous renseigner sur la classe `scrlttr2` qui est très personnalisable.

Pour écrire vos lettres encore plus facilement, vous pouvez même l'écrire en Markdown et utiliser la classe `lettre` pour obtenir votre document PDF.

Vous avez donc toutes les cartes en main pour écrire vos lettres encore plus facilement. Et s'il s'agit d'une lettre de motivation, vous pouvez jeter un coup d'œil à la classe `moderncv` qui vous permettra de créer un joli CV pour aller avec votre lettre.

Je remercie paspro pour ses remarques durant la bêta et Dominus Carnufex pour ses corrections. Merci également à tous ceux qui ont contribué à la bêta. Sans oublier d'applaudir le travail remarquable d'Arius et des autres validateurs.