

Compte rendu des conférences de la Journée GUTenberg .....	2
Compte rendu de l'assemblée générale de l'association GUTenberg .....	16
Les différents travaux de l'association .....	26
Quelques nouvelles de Pologne .....	28
Aperçu du package nicematrix .....	29
Publicité .....	37
De nouveaux packages sur le CTAN .....	38
Utiliser des boucles avec TikZ .....	45
Brèves typographiques .....	50
La fonte de ce numéro : Libertinus .....	52
Comptes rendus de lecture .....	56

Chers adhérents,

ce numéro de la *Lettre* GUTenberg paraît enfin. Vous y trouverez le compte rendu de l'assemblée générale du 17 avril dernier, des articles T<sub>E</sub>Xniques (notamment sur le formidable package *nicematrix*) et d'autres s'intéressant à la typographie d'une manière plus générale, quelques métaphores, un nombre cistercien, une fonte finlandaise du XV<sup>e</sup> siècle, un soupçon de cryptographie... et vraisemblablement quelques coquilles ayant échappé à notre vigilance. Ce dont nous vous prions de nous excuser.

Remarquez-vous combien la Lettre a changé? Nous espérons que non, bien que sa classe ait été profondément remaniée; mais nous avons tenu à garder la mise en page de notre publication. C'est en rédigeant les trois derniers numéros de la Lettre, en les composant, en les compilant que, peu à peu, l'équipe rédactionnelle s'est approprié la Lettre, en a questionné ses us et coutumes et a développé les outils nécessaires aux choix que nous avons faits. Denis Bitouzé en a codé l'essentiel, dans un constant échange avec l'équipe. Et il a rédigé une remarquable documentation qui permettra aux rédacteurs futurs d'apporter aisément leur concours à l'effort collectif. En attendant de la déposer sur le CTAN, ce

## **Avez-vous pensé à régler votre cotisation 2021?**

**Si vous avez oublié, ce n'est ni trop tard ni difficile :**

<http://gutenberg.eu.org/?Adherer-en-ligne>

Voir aussi page 58.

qui la rendra largement disponible et facilement installable, la classe de la Lettre et sa documentation sont disponibles sur le dépôt git de l'association, à l'adresse :

<https://framagit.org/gutenberg/classe-lettre-gut>

Nous encourageons vivement les amateurs de jolis packages d'aller lire le code de la classe de la Lettre. Puisse-t-il susciter des vocations !

Sur ce, nous espérons surtout que le contenu de la présente Lettre vous intéressera. Elle paraît bien tardivement, c'est un fait. Mais pour parvenir aux pages que vous lisez, il nous aura fallu rien moins que 219 commits sur le dépôt git de ce numéro, et 66 sur celui de la classe de la Lettre ! Et depuis le début du mois de juillet, nous sommes conscients du retard pris et de l'urgence qu'il y a à sortir ce numéro ; chaque jour nous y travaillons, et les échanges sont allés bon train au sein de la rédaction. Le numéro sort finalement en ce 12 août, toujours en deux versions <sup>1</sup> ; nous espérons qu'il vous procurera une agréable lecture estivale et que vos réactions seront nombreuses. À bientôt pour la Lettre 45 !

Patrick BIDEAULT

PS1 : notre correspondant de Landerneau nous signale qu'un membre du CA, revanchard et peut-être frustré du peu d'écho rencontré au sein du CA par ses fréquentes et navrantes critiques, a jugé bon de se répandre en contre-vérités dans le *TUGboat*. Pour surprenante que soit cette parution dans un prestigieux organe, dont j'espère que le manque d'exigence rédactionnelle ayant présidé à la publication de cet article ne fut que temporaire, elle confirme la piètre opinion que j'ai de cet administrateur, dont le nom n'est, en ces circonstances, pas même digne d'être cité ici.

PS2 : le secrétaire adjoint de l'association, Maxime Chupin, a publié un article sur l'association dans deux revues de la communauté de la recherche mathématique française, la *Gazette des mathématiciens* et *Matapli* <sup>2</sup>. Autrement plus constructif que celui cité dans le post-scriptum précédent, cet article <sup>3</sup> a été lu avec plaisir par une mathématicienne reconnue, qui a pris la plume pour nous dire tout le bien qu'elle pensait de l'ouvrage de Raymond Sérout, préfacé par Dominique Foata, *Le Petit Livre de T<sub>E</sub>X*, publié en 1989, ré-édité en 1992, dans lequel notre association figure en bonne place. Nous sommes très heureux de la postérité de cet ouvrage, travaillons à la pérennité de l'association et espérons voir ses travaux actuels toujours appréciés dans une trentaine d'années !

## COMPTE RENDU DES CONFÉRENCES DE LA JOURNÉE GUTENBERG

Comme cela avait été annoncé, la Journée GUTenberg a été l'occasion de suivre les conférences données par trois contributeurs « francophones non français » à T<sub>E</sub>X (et logiciels annexes) :

1. *Literate programming* avec Org mode (Fabrice NIESSEN)
2. Nouvelle gestion du format PDF par L<sup>A</sup>T<sub>E</sub>X (Ulrike FISCHER)
3. Courte introduction aux packages TikZducks et TikZlings (SAMCARTER)

---

1. Il s'agit d'une version pour lecture sur écran, dont le fond des pages est très légèrement coloré, et d'une version sur fond blanc pour ceux qui préfèrent imprimer les fichiers avant de les lire. Mais d'autres éléments différencient les deux versions de la Lettre et sont proposés à votre sagacité.

2. Maxime Chupin est rédacteur en chef adjoint de la revue Matapli.  
3. Cet article sera mis en ligne en octobre sur le site de Matapli.

Ceux de nos lecteurs qui n’auraient pu y assister en trouveront ci-après un compte rendu assez détaillé (sauf pour la dernière conférence) et, s’ils souhaitent en savoir davantage, pourront visionner leurs enregistrements, par exemple à l’adresse suivante :

<https://tubedu.org/video-channels/gutenberg/videos>.

## LITERATE PROGRAMMING AVEC ORG MODE

Fabrice NIESSEN, de Leuven en Belgique, est informaticien. Il a participé au stage  $\LaTeX$  à Dunkerque d’abord en tant que stagiaire, ensuite à plusieurs reprises en tant qu’intervenant sur deux aspects d’Emacs : l’édition de documents  $\LaTeX$  et le mode majeur `Org`.

C’est ce second aspect qu’il a développé cette fois-ci en nous donnant un aperçu de certaines des possibilités offertes par `Org mode` :

1. l’édition simple à l’aide d’une syntaxe de type Wiki ou Markdown ;
2. la programmation lettrée qui comprend :
  - (a) l’export en des fichiers notamment `.tex` (avec éventuelle génération du `.pdf` correspondant), `.html`, `.odt` ;
  - (b) l’extraction de blocs de code source et génération de code « entrelacé » ;
3. l’exécution de programmes dans n’importe quel langage au sein même du fichier `Org mode`.

Sa conférence a été articulée selon ces trois aspects.

## Syntaxe

**Directives** où l’on a vu que tout fichier `Org mode` commence par des directives qui sont l’équivalent du préambule d’un fichier `.tex` et qui permettent par exemple de spécifier l’intitulé et l’auteur du document mais aussi, pour l’export, la numérotation ou pas des sections, la présence ou pas d’une table des matières.

Certaines directives, spécifiques à l’export  $\LaTeX$ , permettent de spécifier par exemple la classe (autre que `article`, par défaut) souhaitée.

**Cycle des vues de la structure** où l’on a appris qu’un document `Org mode` peut être structuré en l’équivalent des sections, sous-sections, etc., introduites au moyen des caractères (à placer en début de ligne) `*`, `**`, etc. (ou `[Alt]+[←]`). Une fois cela fait, `Org mode` permet d’aisément (`[⇧]+[←]`) boucler entre 3 vues de la structure :

1. celle de (très) loin, où ne sont affichées que les rubriques de plus haut niveau ;
2. celle de loin, où ne sont affichées que les rubriques, mais toutes les rubriques, quel que soit leur niveau ;
3. celle « normale » où est affichée l’intégralité du contenu du document.

**Mise en forme** où l’on a vu comment, au moyen de simples caractères tels que `*`, `/`, `=`, `~`, etc. (pouvant être masqués), on pouvait mettre du texte en gras, en italique, en *verbatim*, en « code » (en vue d’un export, par exemple `\lstinline` plutôt que `\verb`), etc.

**Listes** où l’on a appris combien il était simple de créer les équivalents des listes `itemize`, `enumerate`, `description` et d’en déplacer les « items » :

- soit horizontalement pour en modifier la profondeur ;
- soit verticalement pour en modifier l’ordre ;

et ce, accompagnés ou pas de leurs descendants.

**Tableaux** dont on a découvert à quel point ils étaient simples à créer, à modifier (par exemple pour déplacer des lignes ou des colonnes), à remplir (en créant des suites de nombres, en insérant des formules permettant des calculs, etc.), à clarifier (en alignant les colonnes après édition), et qu'ils offraient certaines des fonctionnalités pratiques des tableurs. On a pu aussi constater que tout peut être dynamique dans le contenu d'un fichier Org mode, notamment par la possibilité de chaîner les valeurs entre différents tableaux.

## Programmation lettrée

La « programmation lettrée » (*Literate programming*), approche de la programmation préconisée par Donald E. KNUTH, permet « [...]d']obtenir, à partir d'un fichier source, deux représentations : l'une utilisée par un compilateur ou un exécutable, le code "entrelacé", et l'autre lue comme une documentation formatée, qui est dite "tissée" à partir de la source lettrée. » [1]

L'idée de la programmation lettrée est donc la suivante : plutôt que de documenter un fichier de code informatique au moyen de commentaires, créer un fichier :

- pouvant être exporté en une documentation exprimée dans le langage naturel d'un humain ;
- entrelacé de morceaux de codes qui, *in fine*, en seront extraits pour être regroupés dans un ou plusieurs fichiers de code (seulement), exploitables par l'ordinateur.

Ainsi que l'a noté Fabrice, sont donc à distinguer ici deux concepts :

**weave** : exporter le fichier Org mode en entier comme documentation « tissée », formatée pour l'homme, généralement en L<sup>A</sup>T<sub>E</sub>X ou en HTML<sup>4</sup> ;

**tangle** : extraire les blocs de code source et générer le code « entrelacé », formaté pour la machine, pour compilation ou exécution ultérieure ;

avec possibilité de changer l'ordre des blocs de code source.

## Export

Fabrice a commencé par montrer comment exporter tout ou partie d'un document Org mode contenant les éléments vus précédemment :

- en un fichier L<sup>A</sup>T<sub>E</sub>X, éventuellement « publié » c'est-à-dire dont la compilation et éventuellement l'affichage du PDF<sup>5</sup> résultant sont lancés en sous-main ;
- en un fichier HTML, par défaut dépourvu de CSS<sup>6</sup> mais pouvant aisément s'en voir adjoindre, ce dont il a fait la démonstration au travers de deux thèmes HTML qu'il a conçus :
  1. [Bigblow](#) ;
  2. [ReadTheOrg](#).

## Extraction

Pour illustrer le « *tangling* », Fabrice a pris l'exemple d'un fichier Org mode contenant, en plus d'un laïus de *documentation*, un bloc de code L<sup>A</sup>T<sub>E</sub>X, en l'occurrence celui d'un quasi ECM<sup>7</sup>. Au moyen d'une directive passée à ce bloc de code indiquant qu'il devait être extrait et spécifiant dans quel fichier, une commande Org mode d'export a en effet généré ledit

4. *HyperText Markup Language* [anglais] : langage de balises pour l'hypertexte.

5. *Portable Document Format* [anglais] : format de document portable.

6. *Cascading Style Sheets* [anglais] : feuilles de style en cascade.

7. [Exemple Complet Minimal](#).

fichier contenant l'intégralité du bloc de code. Il a ensuite montré qu'il était possible de scinder ce bloc en plusieurs sous-blocs, permettant ainsi de raffiner la *documentation* en insérant des explications entre chacun d'eux. Ceux qui connaissent les fichiers `.dtx` de  $\LaTeX$  voient certainement la similarité des méthodes.

L'extraction se fait *a priori* dans l'ordre chronologique du document mais Fabrice a montré qu'il pouvait en être autrement et que les sous-blocs de code précédents pouvaient être nommés pour être extraits dans un ordre arbitraire. Cela suppose de créer un bloc de code supplémentaire :

- contenant (entre autres) les noms des blocs dans l'ordre souhaité, chacun entre balises spéciales permettant de faire comprendre à `Org mode` qu'ils sont à interpréter comme des références;
- équipé d'une directive supplémentaire indiquant donc à `Org mode` de ne pas extraire verbatim les noms de code entre balises mais de les interpréter comme étant des références et de, à la place, injecter dans le fichier cible le contenu de chacun de ces blocs.

Cette directive a pour nom `noweb`, suggéré par Fabrice aux développeurs de `Org mode` en souvenir du package  $\TeX$  éponyme qu'il utilisait du temps où il utilisait encore directement  $\LaTeX$ .

Fabrice a en outre montré que si, dans le bloc de code contenant ces *références* (noms d'autres blocs entre balises spéciales), un texte (arbitraire) figure devant une telle référence, celui-ci est lors de l'extraction ajouté au début de chaque ligne du bloc *référéncé*.

## Exécution de programmes

Il s'agit ici de faire figurer dans le fichier `Org mode` des blocs de codes, de programmes informatiques dans à peu près n'importe quel langage et de demander l'exécution de ces programmes, leurs résultats pouvant être visibles non seulement dans le *buffer*<sup>8</sup> mais aussi également, si souhaité, dans les fichiers par exemple  $\LaTeX$  ou HTML exportés. Il est en outre possible de chaîner les résultats des programmes de blocs différents, y compris s'ils sont programmés dans des langages différents; cela peut notamment se faire dans un bloc de code  $\LaTeX$  qui peut donc ainsi contenir des résultats de calculs effectués ailleurs dans le fichier `Org mode`, en pouvant choisir localement les valeurs initiales des variables impliquées!

Fabrice a terminé sa conférence par la démonstration d'un bloc de code R permettant la création d'un graphique, qu'il a pu afficher dans le *buffer* et qui, par exemple, pourrait possiblement l'être dans le fichier PDF après export en  $\LaTeX$ .

## Références

- [1] *Programmation lettrée*. In : *Wikipédia*. 12 juin 2021. URL : [https://fr.wikipedia.org/wiki/Programmation\\_lettr%C3%A9e](https://fr.wikipedia.org/wiki/Programmation_lettr%C3%A9e).

## NOUVELLE GESTION DU FORMAT PDF PAR $\LaTeX$

Ulrike FISCHER, de Mönchengladbach en Allemagne, est autrice de plusieurs packages  $\LaTeX$ . Elle est aussi assez connue des dinosaures francophones  $\LaTeX$  puisqu'elle répond très régulièrement à des questions sur le *newsgroup* `fr.comp.text.tex`. Mais elle répond en fait à des questions  $\LaTeX$  un peu partout, à tel point qu'elle a été mise à l'honneur lors de la

8. « Tampon » temporaire contenant une copie de travail d'un fichier.

conférence du TUG<sup>9</sup> en 2015 à Darmstadt, au motif que tout le monde a au moins une fois dans sa vie reçu de l'aide d'Ulrike! Depuis 2 ans et demi, elle est membre de la *L<sup>A</sup>T<sub>E</sub>X team*, ce qui témoigne de son niveau d'expertise. Elle qui a appris le français enfant alors qu'elle vivait en Suisse, a donné sa conférence dans un français tout à fait compréhensible.

Ulrike nous a donc présenté la nouvelle gestion du format PDF par *L<sup>A</sup>T<sub>E</sub>X* dont elle est experte puisqu'elle assure la maintenance et le développement des packages *hyperref*, bien connu de tous, et *tagpdf* qui a été créé pour expérimenter (avec *pdfL<sup>A</sup>T<sub>E</sub>X* et *LuaL<sup>A</sup>T<sub>E</sub>X*) certaines exigences des PDF « accessibles », notamment le *tagging* (marquage).

*L<sup>A</sup>T<sub>E</sub>X* permet de produire des fichiers de sortie dans de nombreux formats, mais le format PDF est actuellement le plus utilisé et le plus important. De façon surprenante, le noyau *L<sup>A</sup>T<sub>E</sub>X* contient très peu de code relatif à PDF et, pour la gestion de besoins basiques tels que les liens hypertextes, la couleur, l'inclusion de graphiques, la rotation des pages à l'écran, etc. il faut recourir à des packages.

Ulrike nous a exposé tous les problèmes que cela engendre :

- les packages en question peuvent entrer en conflit, notamment du point de vue de ce qu'ils font relativement à PDF ;
- un fichier PDF peut être obtenu via de nombreux biais, par exemple *pdfL<sup>A</sup>T<sub>E</sub>X*, *LuaL<sup>A</sup>T<sub>E</sub>X* et *L<sup>A</sup>T<sub>E</sub>X + DVIPS*<sup>10</sup> + *ps2pdf*. Tous ces moteurs et *backends* recourent à des commandes différentes pour faire les mêmes choses ; en outre, les packages qui manipulent le PDF doivent maintenir du code (des fichiers pilotes) propre à chaque moteur. Comme les moteurs et les packages évoluent, cela pose de sérieux problèmes de maintenance.

Cela a certes fonctionné jusqu'ici mais les exigences évoluent :

1. l'impression sur papier n'est plus l'objectif principal (écran, formulaires) ;
2. les normes PDF sont de plus en plus requises, par exemple la norme PDF/A pour l'archivage des documents, exigée par les services d'archives ainsi que par un nombre croissant d'universités ;
3. l'accessibilité devient un enjeu important ;
4. il est de nos jours fréquent que les utilisateurs souhaitent exporter de PDF vers d'autres formats, par exemple HTML ;
5. les métadonnées sont de moins en moins négligées.

Cela a conduit au projet d'envergure (sur 3 ou 4 ans) « *Tagged PDF* », mené par la *L<sup>A</sup>T<sub>E</sub>X team* en collaboration avec la *PDF Association* et Adobe. Ce projet, qui a pour ambition de régler les problèmes vus précédemment, nécessite *beaucoup* plus de code spécifique au PDF dans le noyau *L<sup>A</sup>T<sub>E</sub>X* ainsi que de meilleures interfaces.

Pour cela, l'équipe a développé du code s'attachant à résoudre certains des problèmes évoqués ci-dessus. Ulrike nous en a fait le tour du propriétaire que voici.

## Objectif n° 1 : résolution des conflits

Les conflits visés ici sont ceux entre les commandes propres à chaque moteur ou *backend* alors qu'elles ont le même but. Par exemple pour le moteur *pdfT<sub>E</sub>X*, les commandes sources de conflits sont les primitives `\pdfinfo`, `\pdfcatalog`, `\pdfpageattr`, `\pdfpagesattr`, `\pdfpageresources`.

9. *T<sub>E</sub>X User Group* [anglais] : groupe (international) d'utilisateurs de T<sub>E</sub>X.

10. *DVI-to-PS (translator)* [anglais] : (convertisseur) DVI vers PS.



Pour résoudre ces conflits, la commande `\pdfmanagement_add:nnn` a été créée ; il s'agit d'une commande `expl3`<sup>11</sup> qui est donc *a priori* destinée davantage aux auteurs de *packages* qu'aux auteurs de *documents* (quoiqu'une commande  $\LaTeX$  à eux destinée pourrait être fournie ultérieurement). Cette commande doit désormais être utilisée à la place des commandes ci-dessus. Par exemple au lieu de :

```
1 \pdfinfo{/Title (Titre A)}
```

il est recommandé d'utiliser :

```
1 \pdfmanagement_add:nnn {Info} {Title} {Titre A}
```

En plus d'éviter les conflits, on bénéficie alors d'autres fonctionnalités, par exemple la prévention des doublons dans les métadonnées du fichier PDF.

Cela entraîne toutefois des dommages collatéraux : les auteurs de certains packages doivent adapter leur code, et il en est de même pour certains packages (ou classes) s'appuyant sur les premiers.

Mais les auteurs de *documents* sont aussi concernés : lorsque le code source de ces packages a été retravaillé, la  $\LaTeX$  team a dû faire le choix, nécessairement unique, du codage d'entrée et c'est l'UTF-8 qui a été choisi. En effet, celui-ci présente de nombreux avantages sur les autres ; par exemple, si des caractères accentués figurent en argument de `\section`, le fichier `.toc` va les contenir inchangés tandis que, avec les autres codages d'entrée, il sera truffé de scories (telles que `\IeC {\ ' e}`). Or, dans le cadre d'un *tagged* PDF, les informations telles que les intitulés de sections doivent justement être écrites dans le PDF. Les auteurs de documents qui utiliseront ces packages retravaillés dans des fichiers sources `.tex` utilisant des codages d'entrée autres que l'UTF-8, par exemple `latin1`, s'exposent donc à plus ou moins long terme à des dysfonctionnements.

Du fait de ces nombreux changements et incompatibilités, la  $\LaTeX$  team a mis en place un filet de sécurité : le nouveau code n'est pas encore dans le noyau  $\LaTeX$ , mais dans le package temporaire `pdfmanagement-testphase` à charger et à activer explicitement comme suit :

```
1 \RequirePackage{pdfmanagement-testphase} % Chargement
2 \DeclareDocumentMetadata % Activation de la gestion du PDF
3 {
4   <options>
5 }
6 %
7 \documentclass{...}
```

c'est-à-dire notamment avant `\documentclass`. Si un *backend* non automatiquement détectable par  $\LaTeX$  doit être utilisé, il doit être explicitement déclaré, par exemple ainsi :

11. Reconnaissable à son nom qui contient un ou plusieurs `_` et un `:`. Une telle commande n'est utilisable que si elle est précédée de la bascule `\ExplSyntaxOn` qui modifie le code de catégorie des deux précédents caractères afin qu'ils soient considérés comme des « lettres ». On rétablit le régime habituel au moyen de la bascule `\ExplSyntaxOff`. Entre ces deux bascules (dont on notera la similitude avec `\makeatletter` et `\makeatother`), les espaces et changements de lignes sont ignorés en ce sens qu'ils ne provoquent pas d'espaces dans le document final, ces derniers étant obtenus au moyen de `~`.

```
1 \DeclareDocumentMetadata{backend=dvipdfmx}
```

## Objectif n° 2 : meilleur support des *backends*

Cela nécessite une meilleure abstraction des différents moteurs et *backends*. Par exemple, pour insérer de la couleur dans un PDF, de nombreuses primitives internes propres doivent être mises en jeu, notamment :

- `\pdfcolorstack` avec le moteur pdf $\LaTeX$ ;
- `\pdfextension colorstack` avec le moteur Lua $\LaTeX$ ;
- `\special{color push ...}` avec le *backend* DVIPS;
- `\special{pdfcolorstack ...}` avec le moteur X $\LaTeX$ .

Heureusement pour les utilisateurs finaux, ils n'ont pas à se plonger dans ces arcanes et se contentent d'utiliser une « abstraction » (`\color`) qui s'occupe de ces détails. La  $\LaTeX$  team souhaite offrir davantage de telles abstractions, les intégrer aux noyaux  $\LaTeX$  et ainsi éviter aux auteurs de packages d'avoir à coder et maintenir des fichiers pilotes pour gérer toutes ces primitives propres aux moteurs et *backends*.

L'intégration dans le noyau  $\LaTeX$  de telles abstractions a déjà commencé :

- en février 2020 : `expl3` et `l3backend`;
- au début 2021 : `l3pdf` et `l3color`;
- en mars 2021 : `pdfmanagement-testphase` avec :
  - le `l3backend-testphase` fournissant davantage de commandes pour les *backends* ayant vocation à intégrer le noyau  $\LaTeX$ ;
  - les modules :
    - `l3pdfannot` : annotations, liens hypertextes;
    - `l3pdfxforms` : « *Xobjects* » (sortes d'images dans les fichiers PDF);
    - `l3pdffile` : incorporation de fichiers;
    - `l3pdffield` : formulaires;

dont les commandes fonctionnent dans tous les modules.

La création de ces commandes abstraites n'a rien de banal car les commandes propres à chaque moteur ne diffèrent pas que par leur nom, mais aussi par leurs actions pas toutes identiques. Par exemple, la commande `\pdf_version_gset:n` qui permet de fixer la version du PDF à créer, ne fonctionne actuellement qu'avec pdf $\LaTeX$ , Lua $\LaTeX$  et X $\LaTeX$ , et notamment pas avec DVIPS ou `dvisvgm`.

## Exemples d'utilisation/d'utilité

### Transparence

Le package `transparent`, qui permet d'insérer du texte en transparence (éventuellement à cheval sur plusieurs pages), ne fonctionne qu'avec pdf $\LaTeX$  et Lua $\LaTeX$ . Un nouveau package, `transparent-ltx`, le remplace si `pdfmanagement-testphase` est utilisé.

Ulrike nous a montré le code du premier qui n'est pas très long et qui est modérément complexe<sup>12</sup>. Pour placer des éléments dans `/ExtGState`<sup>13</sup>, il fait usage des commandes primitives de pdf $\LaTeX$  et Lua $\LaTeX$  qui posent problème et ne doivent plus être utilisées.

12. NDLR Selon elle, en tout cas!

13. Dictionnaire PDF *external graphics state* chargé de placer des éléments graphiques sur la page.



Pour nous permettre de comparer, Ulrike nous a également montré le code du deuxième, significativement plus court (environ 25 lignes contre 150) et plus simple : il ne définit que deux commandes publiques mais s'appuie sur une « abstraction », en l'occurrence la commande `\opacity_select:n` définie dans le package expérimental `l3opacity`. Ce dernier est utilisable et fonctionne partiellement avec X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X et DVIPS mais ces moteurs vont nécessiter des adaptations pour assurer une prise en charge complète.

## Couleurs

On peut désormais colorer son document sans devoir charger de package dédié. Parmi les commandes couramment utilisées analogues à celle du package `xcolor`, on trouve :

```

1 % Équivalents de \color (bascule)
2 \color_select:n {<(expression de) couleur>}
3 \color_select:nn {<modèle(s)>} {<valeur(s)>}
4 % Équivalent de \colorlet
5 \color_set:nn {<nom>} {<(expression de) couleur>}
6 \color_set:nnn {<nom>} {<modèle(s)>} {<valeur(s)>}

```

utilisables donc par exemple comme suit :

```

1 \documentclass{article}
2 \begin{document}
3 \ExplSyntaxOn
4 \color_select:n {red} Rouge. \par
5 \color_select:n {cyan!50!magenta!10!yellow} Mélange~ de~ couleurs. \par
6 \color_set:nnn {europablue} {RGB} {0,33,99}
7 \color_select:n {europablue} Bleu~ d'Europe.
8 \ExplSyntaxOff
9 \end{document}

```

On peut aussi créer de nouveaux modèles par la commande `\color_model_new:nnn`, par exemple une « séparation », ce qui permet de faire figurer dans le PDF les informations nécessaires à l'imprimeur pour le choix de l'encre à utiliser (il s'agit de ce qu'on appelle les couleurs d'accompagnement). Jusqu'ici, seul le package `colorspace` de Javier BEZOS fournissait cette fonctionnalité ; désormais, cela est fourni d'emblée (du moins avec l'actuel package `pdfmanagement-testphase`) :

```

1 \RequirePackage{pdfmanagement-testphase}
2 \DeclareDocumentMetadata{uncompress}
3 \documentclass{article}
4 \begin{document}
5 Bla bla...
6
7 \ExplSyntaxOn
8 \color_model_new:nnn {BarTone} {Separation} {
9   name = BarTone~ 555~ GN ,
10  alternative-model = cmyk ,
11  alternative-values = { 0.1, 0.2, 0.3, 0.4 }
12 }
13 \color_select:nn{BarTone}{1} BarTone \par
14 \color_select:nn{BarTone}{0.5} BarTone
15 \ExplSyntaxOff

```

```
16 \end{document}
```

et Ulrike nous a montré que les informations correspondantes se retrouvaient bien dans le fichier PDF <sup>14</sup> :

```
1 6 0 obj
2 [ /Separation /BarTone#20555#20GN /DeviceCMYK 5 0 R ]
3 endobj
```

et :

```
1 /color1 cs 1.0 scn /color1 CS 1.0 SCN
2 0 -11.955 Td [(BarT)83(one)]TJ
3 /color1 cs 0.5 scn /color1 CS 0.5 SCN
4 0 -11.955 Td [(BarT)83(one)]TJ 154.421 -543.96 Td [(1)]TJ
```

## Rotation des pages

Lorsqu'un élément du document est trop large pour tenir dans la largeur de l'empagement, une solution est de le faire pivoter. Dans ce cas, on souhaite que, lors de la visualisation à l'écran, la page contenant cet élément pivote elle aussi. Cela est simple à réaliser, sauf pour les éléments flottants (notamment les figures et les tables) car il est difficile de prévoir les pages sur lesquelles ils se trouveront *in fine* et qui devront (elles seules) être pivotées. Avec pdfL<sup>A</sup>T<sub>E</sub>X, c'est possible mais pas simple ; avec le nouveau code, cela fonctionne d'emblée grâce à la commande `\pdfmanagement_add` <sup>15</sup> :

```
1 \RequirePackage{pdfmanagement-testphase}
2 \DeclareDocumentMetadata{}
3 \documentclass[french]{article}
4 \usepackage[T1]{fontenc}
5 \usepackage{kantlipsum}
6 \usepackage{rotating}
7 \usepackage{babel}
8 \begin{document}
9 \kant[1]
10
11 \begin{table}[t]
12 \centering
13 \fbox{xxxxxx}
14 \caption{Un tableau}
15 \end{table}
16
17 \kant[1-5]
18
19 \begin{sidewaystable}
20 \ExplSyntaxOn
21 \pdfmanagement_add:nnn {ThisPage} {Rotate} {90}
22 \ExplSyntaxOff
23 \begin{tabular}{1}
```

14. Celui-ci étant décompressé, on peut examiner au moins une partie de son contenu en l'ouvrant dans un éditeur de texte.

15. Pour voir la différence, compiler ce fichier avec la ligne 21 décommentée, puis commentée.

```

24 \hline
25 Un tableau trop large ne pouvant être visualisé qu'en format
26 à l'italienne (plus connu sous le nom de mode paysage) \\
27 \hline
28 \end{tabular}
29 \caption{Un autre tableau}
30 \end{sidewaystable}
31 \end{document}
32
33 \kant

```

## Package `hyperref`

À l’occasion de son travail sur la nouvelle gestion du format PDF, Ulrike modifie également `hyperref`. Parmi les modifications qu’elle y a apporté, le remplacement de certains fichiers pilotes propres à certains moteurs (`hpdftex.def`, `hluatex.def`, `hxetex.def`, `hdvips.def`, `pdfmark.def`) par un fichier unique : `hgeneric-testphase.def`. Cela présente de nombreux avantages de maintenance puisque ça n’est plus qu’à un seul endroit que doivent être :

- implémentées les nouveautés ;
- corrigées les erreurs.

Par ailleurs, Ulrike en a profité pour procéder à un certain nombre de changements, par exemple :

- les couleurs des liens hypertextes, qui faisaient l’objet de fréquentes plaintes au motif qu’elles n’étaient pas « belles ». Maintenant, elles le sont davantage ;
- les propriétés des liens hypertextes peuvent être modifiées de façon plus fine. Par exemple, il est possible de masquer les liens de type URL<sup>16</sup> seulement ;
- les formulaires PDF, objets de nombreux bogues, dont le code a été revu et amélioré, à commencer par celui des cases à cocher. Il est désormais possible de les créer avec des *appearances*, c’est-à-dire par exemple des images indiquant leurs différents états ; nous en avons eu la démonstration avec des images d’animaux du package `TikZlings` (dont il est question ci-après).

## Conteneur pour les propriétés des documents

Il est apparu important que les documents contiennent davantage de données sur leurs propres propriétés. Pour cela, ont été ajoutées les commandes suivantes utilisables directement dans  $\LaTeX$  et par les auteurs de packages :

- `\AddToDocumentProperties` ;
- `\ShowDocumentProperties` ;
- `\GetDocumentProperties`.

Un exemple d’utilisation est le suivant<sup>17</sup>.

```

1 \RequirePackage{pdfmanagement-testphase}
2 \DeclareDocumentMetadata{
3   pdfversion = 2.0,
4   pdfstandard = A-1b
5 }

```

16. *Uniform Resource Locator* [anglais] : localisateur uniforme de ressource.

17. Ce code corrige celui présenté par Ulrike, auquel il manquait les « labels » `geometry/` et `top-level/`.

```

6 \documentclass{article}
7 \usepackage{hyperref}
8 \hypersetup{
9   pdfauthor = Ulrike Fischer,
10  pdftitle = Conteneur pour les propriétés du document
11 }
12 % Enregistrement de données supplémentaires
13 \AddToDocumentProperties{geometry}{paper}{A4}
14 \AddToDocumentProperties{topic}{Demo}
15 \begin{document}
16 \ShowDocumentProperties
17 \GetDocumentProperties{geometry/paper}
18 \GetDocumentProperties{top-level/topic}
19 \end{document}

```

## Prise en charge des normes

L'une des normes importantes est PDF/A, déjà mentionnée. Pour implémenter celle-ci, il est nécessaire d'incorporer un profil de couleurs. On trouve sur Internet des fragments de code permettant cela, mais ils sont tous dépendants d'un certain moteur. Le package `pdfmanagement-testphase` en offre une syntaxe unifiée sous deux formes :

- une d'assez bas niveau permettant un contrôle fin mais nécessitant de fournir des instructions PDF ;
- une de plus haut niveau :

```

1 \RequirePackage{pdfmanagement-testphase}
2 \DeclareDocumentMetadata{pdfstandard = A-2b}
3 \documentclass{article}
4 \usepackage{hyperxmp}
5 \usepackage{hyperref}

```

## Réinsertion des liens

Si un fichier PDF est inclus dans un autre au moyen de la commande `\includegraphics`, ses liens hypertextes et plus généralement ses annotations sont perdus car supprimés du fichier principal. Le package `pax` de Heiko OBERDIEK permet de résoudre le problème mais ne fonctionne qu'avec pdfL<sup>A</sup>T<sub>E</sub>X et nécessite une application Java externe s'appuyant sur une librairie désormais obsolète.

À nouveau, Ulrike a profité de son travail sur la nouvelle gestion du format PDF pour également créer un nouveau package, `newpax`, qui règle ces problèmes. Celui-ci fonctionne avec tous les moteurs et s'appuie sur Lua qui est fourni avec toutes les distributions T<sub>E</sub>X. Pour tester cela, on pourra commencer par sauvegarder le code suivant dans un fichier `test-lien.tex` :

```

1 \documentclass{article}
2 \usepackage[paperheight=.5cm,paperwidth=4cm,margin=0cm]{geometry}
3 \pagestyle{empty}
4 \usepackage{hyperref}
5 \begin{document}
6 \href{https://www.gutenberg.eu.org/}{Lien vers Gutenberg}
7 \end{document}

```

le compiler avec le moteur de son choix puis inclure le PDF résultant dans un autre fichier, ainsi :

```

1 \RequirePackage{pdfmanagement-testphase}
2 \DeclareDocumentMetadata{uncompress}
3 \documentclass{article}
4 \usepackage[paperheight=1.25cm,paperwidth=5cm,margin=0cm]{geometry}
5 \pagestyle{empty}
6 \usepackage{hyperref}
7 \usepackage{newpax}
8 \ifluatex
9 % Chargement du code Lua
10 \directlua{require(newpax)}
11 % Écriture des fichiers '.newpax' pour newpax.sty
12 \directlua
13 {
14   newpax.writenewpax(test-lien)
15 }
16 \fi
17 \begin{document}
18 \includegraphics{test-lien}
19
20 \newpaxsetup{addannots=false}
21
22 \includegraphics{test-lien}
23 \end{document}

```

On commencera par compiler ce dernier une première fois au moyen de Lua $\LaTeX$  *nécessairement*, puis ensuite avec le moteur de son choix.

## Commentaires et perspectives

Pour conclure, Ulrike insiste sur le fait que ce projet a certes pour but initial les *tagged PDF* mais qu'il devrait également permettre d'améliorer significativement la gestion du format PDF et ainsi aider bien sûr les auteurs de packages, mais également les auteurs de documents.

Ce code modifie beaucoup de choses. Aussi la  $\LaTeX$  team nous prie-t-elle de, même sur des documents « curieux », le tester en faisant précéder `\documentclass` des commandes :

```

1 \RequirePackage{pdfmanagement-testphase}
2 \DeclareDocumentMetadata{<options>}

```

et de faire remonter tout problème rencontré. Cela a bien sûr pour but de minimiser les bogues pour le jour où ce code sera intégré au noyau  $\LaTeX$ . On pourrait croire que les commentaires ne sont attendus que de ceux qui, avec  $\LaTeX$ , génèrent des fichiers PDF. Mais ce n'est en fait pas le cas car la  $\LaTeX$  team ne veut pas non plus introduire de bogues dans les autres formats de sortie et flux de travail (svg<sup>18</sup>,  $\TeX$ 4ht, etc.).

Le code est loin d'être terminé : pour les formulaires par exemple, seules les cases à cocher ont été traitées et la prochaine étape est la gestion des champs de texte. C'est un processus d'autant plus long qu'il faut également modifier certains packages tiers et en contacter parfois les auteurs. Ce travail ne devrait pas être achevé avant un an ou deux.

18. *Scalable Vector Graphics* [anglais] : graphiques vectoriels extensibles.

Les personnes souhaitant en savoir davantage peuvent consulter la page <https://www.latex-project.org/publications/indexbytopic/pdf/> et consulter les documentations disponibles localement au moyen de :

```
1 texdoc -l pdfmanagement
```

Pour pouvoir tester ce code nouveau, il est bien évidemment nécessaire de disposer d'une distribution  $\text{\TeX}$  récente,  $\text{\TL}$ <sup>19</sup> 2021 (à jour) ou  $\text{\MiKTeX}$ .

## COURTE INTRODUCTION AUX PACKAGES **TikZducks** ET **TikZlings**

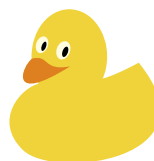
SAMCARTER, qui nous vient aussi d'Allemagne où elle est physicienne des astroparticules, est certainement connue de ceux qui fréquentent le site de questions et réponses anglophone <https://tex.stackexchange.com> sur lequel elle répond fréquemment aux questions posées. Elle est également très présente sur le site <https://topanswers.xyz/tex>. Mais c'est certainement sur le site francophone similaire <https://texnique.fr> qu'on peut la remarquer le plus : elle y intervient fréquemment en apportant des réponses à la fois très sympathiques et très savantes. Elle est tellement incollable sur la classe **beamer** qu'elle en est devenue récemment la principale contributrice. Elle est l'auteur **plusieurs packages** et, lors de cette conférence, elle nous a fait une courte introduction à deux d'entre eux : **TikZducks** et **TikZlings**, fondés sur le package **TikZ**. Comme elle est en français moins à l'aise à l'oral qu'à l'écrit, elle avait pré-enregistré une vidéo que nous avons diffusée et à l'issue de laquelle elle a répondu en anglais aux questions qui lui ont été posées.

Ce qui suit ne couvre que le début de chacune des deux parties de la conférence car celle-ci est avant tout visuelle<sup>20</sup> et le lecteur est invité à en visionner la vidéo, ne serait-ce que pour ne pas manquer les traits d'humour qui l'émaillent.

### TikZducks

Ce package permet de dessiner des canards en caoutchouc, tel que celui de la figure 1 qui

FIGURE 1 – Un canard en caoutchouc



s'obtient on ne peut plus aisément :

```
1 \documentclass{standalone}
2 \usepackage{tikzducks}
3 \begin{document}
4 \begin{tikzpicture}
5   \duck
6 \end{tikzpicture}
7 \end{document}
```

19.  $\text{\TeX}$  Live [anglais].

20. Ne pas manquer les animations finales !




En chargeant la librairie TikZ ducks, on peut aussi en insérer en tant que *short picture* (`pic`) :

```


1 \documentclass{standalone}
2 \usepackage{tikz}
3 \usetikzlibrary{ducks}
4 \begin{document}
5 \begin{tikzpicture}
6   \draw (0,0) pic {duck};
7 \end{tikzpicture}
8 \end{document}

```

Ces canards peuvent être personnalisés de multiples façons, par exemple en modifiant les couleurs des différents éléments qui les constituent <sup>21</sup>.

code	résultat
<pre> 1 \duck[ 2   body=blue, 3   head=green, 4   bill=red, 5   eye=yellow 6 ] </pre>	

Ils peuvent également être ornés d'accessoires :

code	résultat
<pre> 1 \duck[tophat] </pre>	

En plus des options spécifiques à ce package, la commande `\duck` admet toutes celles de TikZ.

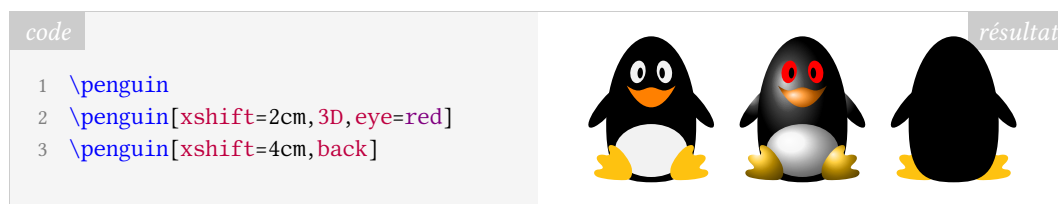
code	résultat
<pre> 1 \duck[tophat=red,rotate=90] </pre>	

## TikZlings

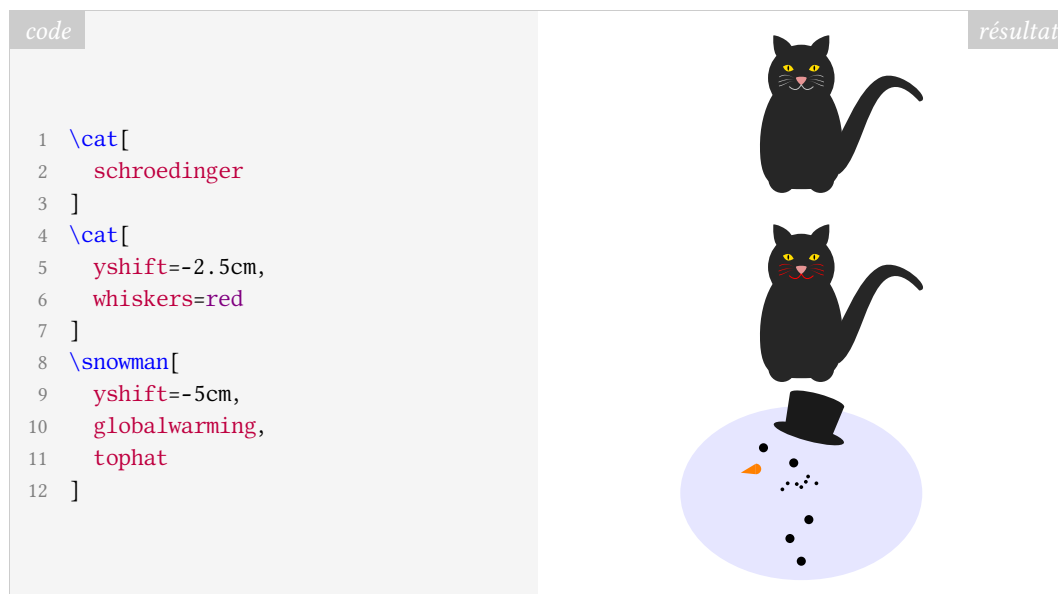
Ce package, « collection de nombreux animaux adorables », fonctionne de façon analogue à TikZducks : il suffit de faire figurer dans un environnement `tikzpicture` la commande portant le nom de l'animal souhaité <sup>22</sup>. Ces commandes acceptent les mêmes options que celles de `\duck` mais sont enrichies de paramètres supplémentaires :

21. Dans la suite, les environnements `tikzpicture` sont sous-entendus.

22. Parmi `\anteater`, `\bear`, `\bee`, `\cat`, `\chicken`, `\coati`, `\elephant`, `\hippo`, `\koala`, `\marmot`, `\mole`, `\mouse`, `\owl`, `\panda`, `\penguin`, `\pig`, `\rhino`, `\sheep`, `\sloth`, `\squirrel`, `\snowman`.



dont certaines propres à certains personnages<sup>23</sup> :



On appréciera l'humour de SAMCARTER et la qualité des solutions techniques qu'elle utilise pour ses personnages. Et on jettera un œil sur son nouveau package [TikZbricks](#)!

Denis BRTOUZÉ

## 🐧 COMPTE RENDU DE L'ASSEMBLÉE GÉNÉRALE DE L'ASSOCIATION GUTENBERG

Le 17 avril 2021 a eu lieu l'assemblée générale de l'association (en visioconférence). Les discussions ont pris place sur la plate-forme BigBlueButton, de vive voix et sur le *chat*, et les votes annoncés à l'ordre du jour ont été menés sur l'outil de sondage de BigBlueButton. Les procurations étaient autorisées et ont été comptabilisées *a posteriori*.

Comme lors de la précédente rencontre entre adhérents, la discussion fut riche, foisonnante de débats et ouverte aux attentes de la communauté : le compte rendu ci-dessous tente de lui être le plus fidèle possible, en respectant l'ordre chronologique, tout en se permettant quelques libertés pour réorganiser *a posteriori* les idées échangées. L'auteur de ces lignes prie par avance les personnes citées de bien vouloir l'excuser en cas de mauvaise retranscription de leurs interventions et est bien entendu tout à fait disposée à en discuter<sup>24</sup> et à corriger ou amender publiquement des points dans la prochaine *Lettre* le cas échéant.

### Ordre du jour

Après les [présentations liées précédemment](#), l'assemblée générale proprement dite est ouverte à 14 heures 45 avec une trentaine de personnes en ligne sur la centaine d'adhérents

23. Attention ! L'option `schroedinger` de la commande `\cat` rend l'état du chat imprévisible.

24. Les commentaires sont les bienvenus à l'adresse [secretariat@gutenberg.eu.org](mailto:secretariat@gutenberg.eu.org).

à jour de leur cotisation. L'ordre du jour était le suivant :

- exposé des bilans moral et financier, discussion, vote de l'approbation des bilans ;
- présentation de la proposition de nouveaux statuts, le projet en cours étant librement consultable à l'adresse :

<https://framagit.org/gutenberg/projet-de-statuts>

- discussion, vote éventuel ;
- sujets divers.

## Bilan moral

Le bilan moral est présenté par Maxime CHUPIN, secrétaire adjoint de l'association. Tout d'abord, du point de vue administratif et statutaire, l'association fait désormais appel à un service de domiciliation chez [www.lestricolores.fr](http://www.lestricolores.fr) et dispose d'une nouvelle adresse :

Association GUTenberg  
15 rue des Halles – BP 74  
75001 Paris

Le courrier est scanné et réadressé une fois par mois. La base de données des adhérents, actuellement sur un système développé par Yannick TANGUY, est en cours de migration vers Dolibarr, qui est un outil plus complet permettant de gérer plus d'aspects administratifs. Pour la trésorerie, le savoir-faire a été transmis à Flora VERN, qui est en train de simplifier les procédures, en particulier avec un changement de banque à venir (pour éviter d'avoir recours à une banque régionale). L'adhésion par virement a été mise en place et est à privilégier (l'IBAN de l'association est rappelé en dernière page de la présente *Lettre*). Enfin, de nouveaux statuts sont en cours de préparation et font l'objet du point suivant de l'ordre du jour.

Au sujet des outils informatiques de l'association, le chantier de migration du serveur a commencé, en étant rigoureusement documenté, en particulier par Yoan TOURNADE. Cette migration était rendue nécessaire car le serveur n'était plus mis à jour et que certains services avaient des failles de sécurité ou étaient hébergés ailleurs (liste [gut@ens.fr](mailto:gut@ens.fr), base de données etc.). C'est d'ailleurs l'occasion de moderniser un certain nombre de services (site *web*, FAQ, ...) ou d'en mettre en place de nouveaux (archivage des listes de diffusion, compilateur en ligne, ...). Un serveur a été loué chez OVHcloud en février (un peu de retard a été pris suite à l'incendie du *datacenter* de Strasbourg). Il fait pour l'instant doublon avec le serveur loué chez Gandi, mais cette situation est temporaire.

Maxime évoque enfin les démarches vers et pour les adhérents, avec tout d'abord la mise en place d'une communication plus importante : trois numéros de la *Lettre GUTenberg* sont parus (électroniquement) depuis novembre, et la refonte de la classe est en cours<sup>25</sup>. Dès que les discussions sur le nouveau nom de domaine auront abouti, une liste de diffusion des adhérents sera également créée, afin de permettre des discussions sur la vie de l'association (la liste [gut@ens.fr](mailto:gut@ens.fr) étant réservée aux discussions plus TeXniques).

La foire aux questions [faq.gutenberg.eu.org](http://faq.gutenberg.eu.org) est un outil précieux. Afin d'en augmenter la visibilité et de faciliter le travail, Denis BROUZÉ a commencé sa refonte avec le système de documentation Sphinx, afin de la moderniser et de permettre les contributions dans d'autres langages que la syntaxe DokuWiki, comme  $\LaTeX$  ou Markdown. Le projet en cours est librement consultable à l'adresse :

<sup>25</sup>. Cette dernière a bien avancé depuis et est en voie d'être achevée, sous l'impulsion de Denis BROUZÉ. L'aspect est resté le même, comme vous pouvez le constater dans le présent numéro.

[dbitouze.gitlab.io/test-faq-fr](https://dbitouze.gitlab.io/test-faq-fr)

La rédaction en chef des *Cahiers* GUTenberg a été reprise en main par Jean-Michel HUFFLEN. Le prochain numéro, comportant 3 contributions originales et 2 traductions est prévu dans les prochains mois<sup>26</sup>. Jean-Michel s'occupe de la production du contenu ; il reste à affiner les aspects concernant l'impression, le financement et la publication en ligne.

Enfin, il est envisagé de proposer de nouveaux moments d'échange pour regrouper les adhérents de l'association (comme celui du 30 janvier, qui a rassemblé en visioconférence une trentaine de personnes avec de nombreux retours positifs), ainsi que des journées GUTenberg avec assemblée générale (comme celle-ci), avec possiblement une AG extraordinaire à l'automne pour valider les nouveaux statuts.

Les statuts précisant qu'il n'y a pas de quorum, le bilan moral est approuvé avec 26 suffrages exprimés, 25 votes oui et 1 vote non.

## Bilan financier

Le bilan financier est présenté par Flora VERN, trésorière de l'association. Après quelques mois de latence, Flora a obtenu les accès aux banques et à Paypal. Le bilan d'exercice est très légèrement excédentaire sur 2020 (de l'ordre de 177 euros, le détail étant paru dans la *Lettre* numéro 43), ce qui montre le support des adhérents (la crise qu'a traversé l'association a relancé les adhésions en 2020, avec 104 adhésions, et à présent 113 adhésions en comptant les adhésions rétroactives). Cet excédent est toutefois insuffisant pour pouvoir prévoir beaucoup d'actions utiles pour l'association. Les nouveaux frais engagés sont la domiciliation et la location du serveur. En contrepartie, des économies sont prévues, en particulier sur les envois postaux. Le bilan 2021 permet de prévoir un excédent de l'ordre de 1500 ou 2000 euros. Le bilan est assez faussé en 2020, en raison de la régularisation des années précédentes (beaucoup de chèques en retard ont été encaissés).

Au niveau du bilan comptable, l'association a environ 30000 euros d'économies. La dette des *Cahiers* représente 22000 euros si l'on décide de rembourser les abonnements et 18000 euros si l'on décide d'envoyer les nouveaux *Cahiers* à paraître. La solution *a priori* retenue est de servir les abonnements grâce aux *Cahiers* à paraître pour garder un fonds de roulement pour les projets de l'association.

Pour les adhésions, il est précisé que celles-ci ont effet pour une année civile.

Les statuts précisant qu'il n'y a pas de quorum, le bilan moral est approuvé à l'unanimité avec 26 suffrages exprimés.

## Projet de statuts

Après les votes, Flora VERN reprend alors la parole pour exposer le projet de nouveaux statuts. Elle rappelle que l'un des principaux objectifs de cette refonte est d'éviter à l'avenir le même genre de crise qu'en 2020, et de donner à l'association les moyens d'organiser une assemblée générale même si le conseil d'administration manque à ses devoirs. Pendant qu'elle travaillait sur ces statuts, elle s'est rendue compte que l'association a des statuts-type qui ne correspondent pas très bien à ses besoins, et qu'il serait peut-être bien de tout revoir pour parvenir à quelque chose de plus adapté.

Rappelons que le projet en cours est librement consultable à l'adresse :

<https://framagit.org/gutenberg/projet-de-statuts>

26. Comme annoncé dans ce numéro de la *Lettre*, le contenu de ce *Cahier* est à présent finalisé.

Nous invitons le lecteur intéressé à le garder sous les yeux pour le consulter en parallèle. Flora n'a pas eu le temps d'intégrer tous les commentaires reçus, la version en cours n'est pas tout à fait identique à la version sur framagit.

Dans le préambule, elle a souhaité garder pour des raisons historiques la phrase initiale expliquant quand l'association a été constituée, tout en précisant que les statuts seront révisés à l'occasion de l'AG extraordinaire de l'automne prochain.

Le premier article concerne la dénomination de l'association. L'idée est de réserver le nom (même si son usage n'est pas commercial). On lui a fait remarquer par mail de modifier Xe~~L~~A~~T~~E~~X~~ en Xe~~T~~E~~X~~ pour harmoniser avec les autres moteurs; cela sera pris en compte.

L'adresse du siège social va être supprimé des statuts. En effet, s'abstenir d'inscrire l'adresse permet ne pas avoir à modifier les statuts en cas de changement d'adresse (l'entreprise de domiciliation peut elle-même être changée ou changer d'adresse). Notons au passage que c'est le conseil d'administration (CA) qui décide de ladite adresse. Ceci n'a pas pour but d'enlever du pouvoir à l'assemblée générale (AG), mais vient du fait qu'un changement d'adresse a rarement lieu par choix, et qu'on souhaite pouvoir le faire vite lorsqu'on doit changer l'hébergement. Elle a rappelé que l'information doit être précisée à la préfecture, aux adhérents et aux tiers via le site *web*.

Arthur ROSENDAHL confirme que c'est très bien de supprimer l'adresse, car c'est juste une décision administrative. François DRUEL propose de faire figurer l'adresse dans le règlement intérieur, mais Arthur rappelle que nous n'avons pas de règlement intérieur. Flora suggère qu'on peut se contenter de la mettre sur le site (il n'y a pas d'obligation de la mettre dans les statuts et il n'y a pas de raison de faire de règlement intérieur dans notre cas). François explique qu'un règlement permet de donner beaucoup de contrôle au CA, ce qui évite de faire voter des détails en AG (tels que des changements d'adresse ou de banque) étant donné que les statuts ne sont pas censés changer. Suite à des remarques d'Arthur et Yoan TOURNADE, Flora confirme qu'il sera bien écrit dans les statuts qu'il faut déclarer les changements d'adresse en préfecture.

Le troisième article porte sur l'objet de l'association. Elle a essayé de les étendre un peu pour nous protéger, ceci afin qu'on ne puisse pas reprocher à l'association d'avoir des activités en dehors des statuts : ce ne sont pas forcément des activités qu'on va faire, mais des activités qu'on se réserve le droit de faire. Dans le *chat*, Arthur rappelle qu'il y a une exigence légale que toute activité commerciale soit inscrite dans les statuts (article 336 du code du commerce), mais ce sera *a priori* plutôt dans l'article 4. Jérémy JUST rebondit sur cette remarque pour expliquer que même si cet article 3 peut paraître être un paragraphe un peu fourre-tout, il constitue l'essence même de l'association. Il appelle à ne pas hésiter à faire des critiques dessus pour faire figurer toutes les choses qu'on veut faire ensemble. Il faut éviter que l'association mène des actions qui sont effectuées en dehors de son objet.

Flora en profite pour rappeler que l'objet (article 3) concerne plutôt les idées et les objectifs, et les activités (article 4) les moyens qui permettent de les réaliser. Elle insiste sur le fait d'indiquer les manques car l'association ne peut pas faire plus que son objet social ou les moyens qu'elle a décidé d'utiliser. Ce sont des éléments qui doivent être publiés en préfecture. Elle précise (simple rappel à la loi) que l'association poursuit un but non lucratif et rappelle les devoirs qui y sont liés.

Flora demande alors s'il y a d'autres idées dans l'assemblée. François DRUEL propose d'ajouter un sixième point au sujet de « toute autre activité se rapportant aux objets précédents », de façon à pouvoir inclure tout ce à quoi on n'aurait pas pensé. En prenant exemple sur d'autres statuts, il propose la formulation « toute opération de quelque nature et importance qu'elle soit dès lors qu'elle peut concourir ou faciliter la réalisation des activités

visées aux alinéas sus-précédents ». Bastien DUMONT propose d'ajouter (dans les objets et les activités) l'aide au développement de logiciels T<sub>E</sub>X (par exemple en proposant, comme c'est déjà le cas, de créer un dépôt sur le serveur Git de l'association). Jérémy remarque que cela rentre dans le paragraphe 2 des activités : Flora pourra rajouter une précision dans ce paragraphe. Bastien note que cela manque dans les objets, mais Philippe ARMINGAUD pense que des objets correspondent (promouvoir en 2 et favoriser les échanges en 3). Tout ceci est donc à préciser. Yvon HENEL demande enfin si l'article 4.4 englobe l'écriture de code, ce à quoi Flora répond par l'affirmative. Finalement, Patrick BIDEAULT propose de préciser que le soutien est moral, humain ou financier autant dans les articles 4.2 que 4.3.

L'article 5 concerne l'adhésion à l'association (en séparant les membres actifs et les membres d'honneur). Flora rappelle qu'on va ajouter que l'adhésion concerne l'année civile. C'est l'AG qui fixe le montant sur proposition du CA (car l'AG ne peut voter que par oui ou par non). Flora rappelle les tarifs qu'on a aujourd'hui. Les personnes morales sont essentiellement des laboratoires, qui payent un peu plus cher mais peuvent rattacher sept personnes. Jérémy rappelle que le seul changement est qu'avant, pour les entreprises ou laboratoires, la précision était par site, pour éviter que le CNRS adhère une seule fois par exemple, en faisant adhérer sept laboratoires. Il pense que la précision n'est plus utile mais se pose la question. François DRUEL ajoute qu'on peut aussi préciser qu'une adhésion de laboratoire comporte un nombre maximum de sièges à l'assemblée générale, Flora rappelle que c'est actuellement sept maximum. Mais Arthur remarque que cela ne répond pas au problème soulevé par Jérémy (sept personnes sur un même site). Flora dit qu'elle peut préciser qu'il s'agit de sept personnes physiques sur un même site. Flora a préféré garder l'article 5.2 sur les membres d'honneur. Jérémy rappelle que certains sont abonnés aux *Cahiers* pour les 30 ans à venir donc c'est bien de l'avoir remis en évidence. Flora a revu les conditions pour être membre d'honneur et se demande si cela doit bénéficier à la communauté francophone ou non (par exemple Donald KNUTH). C'est un réabonnement automatique, les membres d'honneur doivent préciser quand ils ne veulent plus être abonnés. Denis BITOUZÉ demande si les membres d'honneur ont droit de vote. François signale que cela dépend de ce qu'on veut faire avec les membres d'honneur. Dans certaines associations, c'est simplement du *name dropping*, ils sont dispensés de cotisation mais n'ont pas droit de vote, ce qui permettrait par exemple de nommer Donald KNUTH membre d'honneur de notre association. Cela permet également d'honorer à titre posthume. Jérémy précise que non, car les adhérents sont radiés automatiquement à leur décès. Il faudrait modifier l'article sur la radiation et se poser la question du droit de vote.

Le sixième article traite justement de la perte de la qualité de membre. Sur demande de Philippe MANÉ, Flora a précisé ce qu'était un motif grave dans ce cadre pour éviter des dérives de la part du CA. Elle décrit la procédure. François, rappelant que lorsqu'on en arrive à de telles extrémités les relations sont généralement mauvaises entre le membre et l'association, propose de remplacer la phrase sur le remboursement au *pro rata* par « les cotisations restent acquises dans tous les cas » pour simplifier les choses. Arthur a l'avis opposé, en proposant de lui rendre l'argent, quitte à en perdre un peu. Flora avait mis le remboursement pour les cas d'injustice (radiation proposée par le CA et refusée par l'AG). Yannick TANGUY précise ce qui se passe dans l'assurance et propose également de proposer le remboursement systématiquement. Emmanuel PABION rappelle de mettre à jour la fin de l'article (sur la perte de la qualité de membre en cas de décès) en fonction de la discussion sur la qualité de membre d'honneur à titre posthume. Flora rappelle également qu'un adhérent a jusqu'au 31 décembre de l'année en cours pour régulariser son adhésion. Christophe POULAIN rappelle à ce sujet qu'il faut bien définir les dates et durées d'adhésion et Patrick BIDEAULT confirme qu'il s'agit bien d'adhésions par année civile.

Le septième article rappelle les relations avec le TUG. Un adhérent a rappelé à Flora qu'on



n'est pas affiliés avec le TUG (pas une fédération), mais qu'il y a des liens, ce qui permet par exemple à l'association de proposer une adhésion conjointe avec tarif préférentiel. Jérémy s'interroge sur ce point, pour savoir si l'on a obligation de le proposer ; Flora répond que nous avons l'obligation de négocier les tarifs avec eux, mais pas de proposer le tarif, s'il n'est pas intéressant.

Le huitième article concerne l'assemblée générale. C'est l'instance supérieure de l'association. Le but était de clarifier ce passage pour éviter les crises de gestion. Yannick TANGUY propose de préciser le terme de secrétariat, jamais défini au préalable dans les statuts. Flora explique que les fonctions sont définies plus loin et que les président et vice-président ont le même rôle, de même que les secrétaire et secrétaire adjoint et que cela évite des lourdeurs dans la rédaction d'omettre cette précision. Flora propose de fixer une date butoir (par exemple le dernier samedi du mois de septembre) pour l'assemblée générale par défaut. Jérémy propose de fixer cette date plus tard dans l'année (si jamais une date n'a pas été trouvée d'ici là). Dans tous les cas, l'idée est de forcer le CA à convoquer une AG tous les ans (Patrick rappelant que la date normale d'une AG est plutôt en début d'année, pour faire le bilan financier de l'année écoulée). François DRUEL rappelle de dire qui préside dans ce cas une telle AG de plein droit, par exemple le doyen d'âge (des présents) de l'association. Jérémy propose que les statuts définissent également les modalités (aspect matériel, lieu ou électronique). François suggère qu'on peut proposer que le doyen d'âge convoque l'AG. Flora ajoute qu'à défaut ce soit le deuxième plus âgé. Arthur dit que c'est bien de prévoir ce qui se passe en cas de vacance (AG au plus tard fin novembre), mais il faut rappeler qu'en principe c'est le président élu qui convoque et préside. Flora propose de rassembler tout cela dans un point 8.4 dérogatoire, pour une AG en cas de vacance. Jean-Michel BOUCART signale qu'il manque la majuscule à « Assemblée » et demande de préciser s'il y a des AG extraordinaires.

Jérémy rappelle qu'avant, pour les AG régulières, les convocations et ordres du jour devaient être envoyés au plus tard 15 jours avant la date de l'AG, ce qui était un minimum pour que les gens puissent s'organiser. Cette limite a disparu mais va être remise dans la nouvelle version. Il rappelle que par le passé les gens auraient préféré trois mois (plutôt qu'un mois et demi). François demande s'il y a un minimum (pour les AG de copropriétaires, c'est trois mois), mais Arthur rappelle que pour les associations loi 1901, on n'est même pas obligés d'avoir d'AG, donc il n'y a pas de minimum. Jérémy signale toutefois qu'il faut fixer un minimum pour préparer l'ordre du jour, et qu'en général la date est connue bien en avance afin de pouvoir s'organiser. Patrick BIDEAULT demande alors s'il est possible de dissocier la convocation de l'envoi de l'ordre du jour. C'est généralement l'ordre du jour qui est compliqué, car la date est souvent connue un mois et demi à l'avance. Il propose donc de noter un mois pour la date et 15 jours pour l'ordre du jour. Christophe POULAIN demande si la modification des statuts est prévue en AG classique (comme il l'a vu dans l'article 8.3). Flora pense que ce sera en AG extraordinaire pour cette fois-ci.

Patrick signale également qu'autant il est normal de prévoir les votes à l'avance, mais qu'on peut également prévoir d'autoriser d'aborder un sujet sans l'avoir prévu 15 jours à l'avance. François rappelle que les *points divers* permettent d'ajouter des points à l'ordre du jour, avec des modalités, par exemple sur proposition d'un des adhérents, puis sur acceptation. Comme le dit Emmanuel PABION, cela pose pas mal de problèmes, mais on peut sans doute s'arranger. Christophe dit qu'on peut également clôturer l'AG puis passer aux points à aborder lors de l'AG suivante. Comme il y a un délai entre la convocation et l'envoi de l'ordre du jour, Jérémy signale qu'on peut en profiter pour dire aux adhérents qu'ils peuvent soumettre des points à l'ordre du jour. Patrick rappelle que ceci est quelque chose à encadrer, puis Flora suggère que le CA réponde qu'on peut mettre la question à l'ordre du jour, mais qu'on la soumet au vote avant de la traiter. Ceci n'a pas besoin d'être inscrit à l'ordre du

jour. Denis remarque que ce n'est pas au CA de juger de l'absurdité de la question, donc que toute question doit alors être soumise au vote. Ces modalités restent donc à affiner.

Enfin, Flora rappelle qu'il n'y a pas de *quorum*, les décisions étant prises à la majorité des membres présents. Patrick fait la remarque au passage qu'on est en train d'établir un document complexe et qu'il a beaucoup apprécié la version amendée de Flora. Il se demande s'il ne faudrait pas laisser sur le site une version commentée des statuts, afin que nos successeurs comprennent d'où ils viennent et le procédé de réflexion qui y a mené. Il s'agirait juste d'un document de travail à destination des gens qui s'intéressent aux statuts.

Si l'on aborde le sujet des votes, Jérémy dit que cela fait maintenant un an qu'on fait des votes électroniques. Il fait remarquer que nous ouvrons les votes à l'avance afin que les absents puissent voter avant l'AG et que nous nous passons de procurations (soit techniquement impossibles, soit difficiles à gérer). Cela fait que des gens votent sans avoir eu les débats. Il propose donc d'ouvrir les votes après l'AG (pendant une durée telle que 48 heures), afin que les gens aient connaissance des débats. En conséquence, le résultat des votes ne serait pas connu pendant l'AG. Flora dit qu'il faudrait alors envoyer le compte rendu avant l'expiration de ce délai. Jérémy dit qu'il faudrait envoyer le compte rendu des débats, effectivement. Patrick considère cette solution ingérable et rappelle que c'est pareil avec les procurations, le mandant donne ses instructions au mandataire à l'avance. Céline CHEVALIER s'interroge sur le cas où un point dépend d'un point voté pendant l'AG et ne peut donc pas être traité sans les résultats du vote, Flora rebondit sur cette remarque. Christophe dit qu'en étant prévenus un mois avant, il n'y a pas besoin de laisser les votes ouverts après l'AG. François signale également la problématique du fait que c'est l'AG qui décide, et Patrick dit qu'effectivement c'est la notion même d'AG qui est remise en cause par un tel projet.

Au sujet des procurations, Flora rappelle qu'elles sont systématiquement données aux membres du CA, ce qui n'est pas démocratique non plus. On pourrait donc autoriser les procurations, mais dans la limite de deux pouvoirs par personne. François a déjà vu des cas où l'on répartit les dizaines de procurations aux présents, mais la difficulté est alors de compter les votes. Flora répond que les outils électroniques organisent l'anonymat donc il n'est pas facile de compter les procurations. Denis BITOUZÉ rappelle que le vote d'aujourd'hui est électronique, mais permet de comptabiliser les procurations (manuellement). Flora s'interroge sur le fait que cela oblige à voter deux fois la même chose, mais Denis lui répond qu'on peut préciser qu'on veut voter autre chose, manuellement, après coup. Dans un vote à bulletin secret, ce n'est pas faisable. Flora dit qu'avec Belenios, on peut donner deux codes à la personne. Céline fait remarquer que si celui qui a donné des procurations donne des consignes, il peut tout aussi bien voter en avance. Jérémy remarque qu'un membre peut donner des consignes en fonction des débats. Christophe dit que le vote est ouvert avant les débats, et qu'on peut y ajouter les procurations pour ceux qui préfèrent. Jérémy dit qu'il faut alors faire confiance aux gens de ne pas voter deux fois. François clôt le débat en proposant de créer un groupe de travail pour étudier les solutions de vote en ligne et accepte la proposition de Denis d'en devenir responsable. Il regardera aussi le « jugement majoritaire ». Philippe Mané est aussi volontaire pour faire partie de ce groupe<sup>27</sup>.

Dans les remarques diverses, Jean-Michel BOUCART signale que la révocation d'un membre du CA semble étrange et demande qui la propose et sous quel motif. Emmanuel PABION demande si la mention des vidéoconférences ne risque pas de limiter vis à vis des évolutions technologiques possibles. Flora rajoutera que c'est l'AG *extraordinaire* qui peut modifier les statuts. Arthur demande si l'on veut vraiment interdire à l'AG ordinaire de le faire. Jérémy dit que c'est une garantie de stabilité des statuts. Flora propose de quantifier

27. Nous allons mettre en place une liste de discussion pour ce projet, n'hésitez pas à vous manifester auprès du secrétariat si vous souhaitez y être inscrit.

(modification de plus de 3 articles), mais Jérémy estime qu'il faut rester simple. Ces points restent encore à décider.

Le neuvième article traite du conseil d'administration. Flora a ajouté des détails (comme la durée maximale de 12 ans au CA) pour forcer que les mandats tournent, même si le but n'est pas de limiter les gens qui voudraient s'investir dans l'association. Patrick renchérit en insistant sur le fait que si en 12 ans on n'a pas trouvé de successeur à un poste, il y a un problème. Cela pousse la communauté à s'investir. Flora rappelle les règles du renouvellement par moitié : l'idée est de garder une élection par moitié tous les deux ans, peu importe s'il y a eu des changements entre temps. Christophe POULAIN demande qui décide de la composition de la moitié : Arthur répond que lors de la fondation ils avaient tiré au sort, Flora dit qu'il faudra prévoir une disposition transitoire et Patrick rappelle que le bureau actuel a dit qu'il remettra ses mandats actuels en jeu à l'assemblée 2022 et il s'y tiendra. Mais Flora signale qu'on ne sait pas ce que feront les autres membres et qu'on pourra donc prévoir un tirage au sort, Patrick acquiesce : effectivement, s'il y a des gens qui démissionnent, cela fera plus de postes vacants.

Flora rappelle que l'idée du renouvellement par moitié a été proposé car cela permet d'avoir une continuité. Mais il complique la vie, elle se demande donc s'il s'impose. Denis répond qu'il n'y est pas attaché. L'idée de Patrick était surtout de garantir une certaine stabilité. Jérémy rappelle que ça a plutôt été un renouvellement par tiers jusqu'ici, et qu'il n'y a jamais eu besoin d'imposer le renouvellement par moitié. Il trouve que la solution proposée est bien et évite des pertes de savoir-faire, avec une association qui ne fonctionnerait plus pendant un an. Il estime qu'en imposant que les gens se représentent s'ils veulent renouveler leur mandat, il est très probable que cela se passe naturellement. Flora dit que nous pouvons supprimer le renouvellement par moitié, dire que les mandats font deux ans, que les gens peuvent se représenter, et renforcer la période de transition (elle a mis deux mois actuellement, on peut monter à trois ou quatre mois). François est d'accord avec la lourdeur du remplacement par moitié ; il propose une autre façon de favoriser le renouvellement, en mettant des mandats de deux ans, reconductibles deux fois consécutives maximum. Christophe signale qu'il peut y avoir renouvellement complet au bout du premier mandat. Patrick trouve la suggestion de François intéressante. Ce qui compte, c'est de penser aux successeurs (ce qui est fait actuellement avec la documentation pour le serveur), et se demande s'il ne faudrait pas mettre une phrase dessus dans les statuts en disant que la transition est prévue et que les anciens membres du CA doivent faire un passage d'information, car ceci ne peut pas être non-dit. Flora y vient, elle en parle plus loin. Yves AKAKPO trouve la durée de deux ans courte, et Christophe estime que c'est long si rien n'est fait. Bastien DUMONT trouve que cela peut rassurer des primo-candidats potentiels. Patrick lui répond que c'est ce à quoi on s'est engagés au sein de la liste GUT-Renouveau, et que ce n'est pas si mal, car cela oblige à une certaine efficacité, un certain dévouement.

Dans l'article 9.1, Flora rappelle les modalités pour les élections : en particulier, on peut se présenter sous forme individuelle ou en liste, mais on vote par candidat. Les candidats sont élus à la majorité absolue et par classement : les 12 premiers seront élus, et s'il y a égalité des voix, on donne la préférence au candidat le plus jeune. Si le nombre de candidats ainsi élus ne permet pas de composer un bureau (de 3 personnes), les suivants seront élus, histoire de toujours avoir un bureau qui tourne. Cela permet d'inciter les gens à candidater au bureau. Elle revient également sur la période de transition (ici de deux mois, à porter à six mois) pour permettre à l'ancien CA de réaliser tous les actes urgents, mais aussi à former le nouveau CA. Patrick trouve que l'emploi du mot formation est très juste.

L'article 9.2 précise la composition du bureau, avec président, secrétaire et trésorier. Il ne pourra y avoir 12 personnes au CA que s'il y en a 6 au bureau, ceci pour éviter d'avoir un

CA trop pléthorique. Flora rajoutera dans cet article que tous les adjoints<sup>28</sup> ont les mêmes pouvoirs que les titulaires.

Flora rappelle également qu'on peut déléguer des tâches à d'autres adhérents, par exemple à ceux qui n'ont pas été élus. Ils ne prennent pas part aux décisions du CA, mais en échange c'est le CA qui est responsable devant l'AG. Il est également possible de déléguer des pouvoirs à d'autres personnes du CA. Denis propose que ce soit le moment de discuter de la proposition d'élargir le CA à 16 membres, mais Flora répond que c'est déjà compliqué de se mettre d'accord à 12 (par exemple au sujet de l'écriture inclusive). Pour revenir au sujet de la délégation, Patrick rappelle que cela fonctionne déjà avec le numéro 13 du CA, Yannick TANGUY, qui s'occupe de la base de données des adhérents. Effectivement, cela n'a jamais été interdit par les statuts, mais le noter permet de formaliser le fait que tout le monde peut s'investir.

Suivent quelques articles à des fins juridico-administrativo-fiscales<sup>29</sup> sur les ressources (article 10), les productions (article 11) et les publications (article 12). Il n'y a pas de cession de droit, mais l'association exige que la diffusion puisse s'effectuer sous licence ouverte. On ne s'oblige pas à être *maintenir* mais on s'autorise à l'être.

Les publications sont les *Lettres* et les *Cahiers*. L'idée est d'avoir un tarif préférentiel pour les adhérents (souscriptions ou abonnements, auxquels Flora n'est pas très favorable, mais Jean-Michel HUFFLEN, qui a repris les *Cahiers*, aura son mot à dire, il faudra en discuter). François dit que cela pose la question du financement de ces publications : est-ce l'association qui joue le rôle d'éditeur et avance les frais, ou bien attend-on des souscriptions pour avoir la trésorerie ? Flora propose de fonctionner par souscriptions pour financer les *Cahiers*, en évitant les abonnements pour éviter de potentielles dettes. Les détails sont à affiner, mais les *Cahiers* seront probablement mis à disposition en ligne moins de deux ans après leur publication papier. À cause d'une formulation dans l'article 12 (« qui est une revue d'information sur les activités de l'association »), Christophe demande si la *Lettre* ne contiendra que les activités de l'association. Patrick et Arthur répondent que cette mention n'interdit rien, et Flora a noté de le préciser dans la nouvelle version.

En réponse à une question de François, il est précisé que les *Cahiers* et la *Lettre* ont un dépôt légal auprès de la BNF (la *Lettre* électronique dispose également d'un ISSN). Patrick en profite pour annoncer que Maxime et lui-même ont le projet de mettre à disposition la *Lettre* en ligne sous une meilleure forme (avec accès aux articles un par un, disponibles en HTML). Emmanuel PABION et François réagissent positivement à cette nouvelle.

Enfin, Flora a réduit l'article 13 sur les listes de diffusion pour ne pas qu'on soit enfermés, le détail pourra être précisé dans le règlement intérieur. Il y a également un petit article 14 sur la dissolution qui prévoit une AG extraordinaire avec les modalités. Elle en termine, pile à l'heure, à 17 heures 30.

Nous tenons à remercier tous ceux qui ont participé à la discussion ou envoyé des commentaires avant l'AG. Il n'est pas recommandé de modifier directement le projet sur le framagit, mais il est possible d'ouvrir des *issues*. Le plus simple est sinon d'écrire par mail au secrétariat. Jérémie demande en particulier des retours pour les articles 12 et 13, afin de

---

28. Une discussion a eu lieu pour choisir entre « vice- », « adjoint » et « suppléant » : « adjoint » a le mérite de signifier qu'on peut se répartir les tâches alors que « suppléant » sous-entend que la personne n'est là que lorsque le membre principal est absent ? Le problème de la féminisation a également été abordé. Au final, il a été proposé de garder « vice-président » mais de parler d'adjoint pour les deux autres postes. Yannick TANGUY rappelle qu'il faudrait remplacer le terme « secrétariat » vu plus haut par « secrétaire ». La cohérence tout au long des statuts sera vérifiée.

29. Même si François rappelle que ces articles ne sont pas neutres, Flora voulait surtout dire par là que ces articles prêteront probablement moins à discussion... et que l'heure tourne.

savoir ce que les futurs auteurs attendent de l'association GUTenberg pour les aider au mieux dans leurs productions.

## Sujets divers

Jean-Michel HUFFLEN prend la parole pour donner des nouvelles des *Cahiers* : sur le modèle du groupe italien, il compte monter un dossier pour être reconnu comme revue de vulgarisation, de façon à ce que les gens qui publient ne le fassent pas que pour le geste (cela fait sans doute partie des choses qui ont coulé les *Cahiers* les dernières années).

Suite aux discussions de janvier, il rappelle que nous aimerions publier plus de numéros hors-séries, livrets ou tutoriels, mais qu'il faudra motiver des auteurs. Il sera certainement plus facile d'être plus crédibles après un certain nombre de numéros de nouveaux *Cahiers*. Dans les deux ou trois prochaines années, Jean-Michel pense que la norme pour les *Cahiers* doit rester le papier, car le tout-électronique a ses défauts, mais qu'il y aura peut-être une transition, dans quelques années.

Patrick BIDEAULT demande à l'assemblée s'il y a des sujets divers à aborder. Rebondissant sur une remarque précédente de Jean-Michel, Philippe MANÉ signale dans le *chat* que Pluton n'est pas une planète naine, ce à quoi Jean-Michel répond qu'en tout cas, ce n'est pas une planète<sup>30</sup>. Denis BITOUZÉ tente de stopper le débat en disant que cela revient à la discussion sur l'écriture inclusive. Patrick conclut en proposant que l'assemblée peut statuer sur le fait que Pluton est un objet céleste et demande si cela fait consensus. Il propose également que Pluton soit mentionné dans le compte rendu de cette AG, voilà qui est fait<sup>31</sup> ! Il demande néanmoins s'il y a d'autres sujets divers à part Pluton.

Pendant ce temps-là, dans le *chat*, Emmanuel PABION a signalé être intéressé par les liens entre sphinx et pandoc. Denis BITOUZÉ prend alors la parole pour lui répondre qu'il n'y a pas de lien direct. En réalité, sphinx demande d'écrire dans le format restructured text. Pour ne pas être obligé d'éditer dans ce format, pandoc permettrait d'écrire sous un autre format et de convertir ensuite.

Jean-Michel revient alors sur le sujet de l'écriture inclusive, et pense que c'est quelque chose qui se fera à son avis en 2022. Il pourra probablement sortir un numéro des *Cahiers* avec un certain nombre d'historiques (écriture épïcène, etc.), mais c'est un projet très lointain. Nul doute que cela assurerait un certain renom aux *Cahiers*, d'après Patrick ; des souscriptions, d'après Jérémy ; et un certain intérêt, de la part de François, qui signale que cela soulève des questions intéressantes de typographie, par exemple comment inclure de l'écriture inclusive en  $\LaTeX$ . Jean-Michel pense que ce serait intéressant de le placer dans une perspective générale, pas uniquement française : il part alors avec Emmanuel PABION dans une discussion germanophone sur le *chat* à ce sujet, nul doute qu'elle sera poursuivie à l'avenir.

Sur cette sympathique conclusion qui apporte un peu de légèreté après la discussion technique des statuts qui lui a précédé, la séance est levée à 17 heures 42 avec les remerciements de plusieurs adhérents envers « le groupe qui s'est retroussé les manches et a relancé l'activité de l'association », pour reprendre les mots de Bruno PIGUET. Merci à vous pour vos retours, c'est un plaisir de sentir l'association revivre.

Céline CHEVALIER

---

30. Jean-Michel avait dit accepter l'idée qu'un article annonçant la découverte d'une nouvelle planète naine soit plus valorisant qu'un article dans les *Cahiers*, mais que cela ne rendait pas nulle la valeur des articles desdits *Cahiers*.

31. Mais peut-être pas dans les statuts, comme Flora l'a proposé en plaisantant. Guilhem BOURRIÉ dit que si l'on mentionne Pluton, il faut associer Proserpine et Cérès. Jean-Michel ajoute alors Makémaké à la liste des planètes naines à citer dans ce compte rendu.

## LES DIFFÉRENTS TRAVAUX DE L'ASSOCIATION

De nombreux chantiers occupent l'équipe ; on notera que le défaut d'ambition ne nous affecte guère. Voici un état des lieux :

### La refonte des statuts

Vous aurez lu, dans le compte-rendu de l'assemblée générale 2021, qui eut lieu en ligne le 17 avril dernier, un résumé des discussions qui eurent lieu à ce propos. Ce travail est toujours en cours. De nombreuses réflexions sont apparues et nous vous invitons à les discuter. Nous prévoyons de parvenir à une version définitive dans le courant de l'automne, de manière à voter l'adoption de ces nouveaux statuts lors d'une assemblée générale extraordinaire qui leur serait dédiée ; elle devrait avoir lieu d'ici la fin de l'année, notamment pour dissocier le vote sur les nouveaux statuts du vote sur le renouvellement du bureau, qui aura lieu lors de l'assemblée générale ordinaire du printemps 2022. Le bureau actuel remettra ses mandats en jeu, comme il l'a promis lors de son élection en novembre 2020.

Le projet de refonte des statuts est disponible dans notre dépôt git, à l'adresse :

<https://framagit.org/gutenberg/projet-de-statuts>

### La FAQ

L'ambitieux travail de Denis Bitouzé, qui consiste à refondre les différentes FAQs actuelles en une ressource fiable et aisément amendable par les utilisateurs, a un peu marqué le pas, et ce tant pour des raisons logicielles (attente d'une extension des fonctionnalités du package `scontents` ; la lecture de la documentation de ce package ouvre des horizons !) que par la formidable activité que Denis a déployée au sein de l'association, notamment pour refondre la classe utilisée pour composer la présente Lettre.

Sur ce, le travail de Denis est accessible ici :

<https://dbitouze.gitlab.io/test-faq-fr/index.html>

### Notre nouveau serveur

Notre nouveau serveur, Leslie, est en route. Divers services sont déjà actifs :

- notre site web est migré, qui plus est sur une version à jour de Spip, le système de gestion de contenu que nous utilisons ;
- un système de sauvegarde est installé ;
- si certains des adhérents ont sur les serveurs qu'ils utilisent de l'espace inutilisé, il leur est possible de le mettre à la disposition de l'association, qui l'utiliserait pour des sauvegardes ; la procédure est aisée ; veuillez contacter le secrétariat si vous souhaitez participer à cet effort collectif : Yoan TOURNADE vous fournira les outils nécessaires ;
- la base de données des adhérents, telle que développée par Yannick TANGUY, est migrée et fonctionne ;
- Dolibarr, le logiciel de gestion interne de l'association, est installé (voir ci-après).

Je rappelle que ces différents outils sont documentés.

Reste notamment à migrer Mailman, le logiciel de gestion de listes de diffusion, et Dokuwiki, le logiciel qui gère actuellement la FAQ.

Évidemment, dès la mise en service de la nouvelle FAQ, des moyens de contributions plus conviviaux seront mis à la disposition des contributeurs.



## Un nouveau nom de domaine

Sachant que gut.org, gut.fr et gutenbergr.fr ne sont pas disponibles, les idées suivantes ont été émises :

- gut.social
- asso-gut.org
- asso-gutenberg.org
- gut-fr.org
- gutexberg.org
- tug-fr.org
- francogut.org
- francotug.org
- gutemberg.org
- gutex.space
- groupetex.org
- grtex.org
- grutex.org
- associationgutenberg.fr

Les adhérents seront consultés à ce sujet<sup>32</sup>.

## La liste de discussions des adhérents

L'année 2020 aura montré à chacun combien les échanges de la liste GUT doivent être indépendants des discussions internes à l'association<sup>33</sup>. Cette liste ouvrira dans l'automne.

## Le logiciel de gestion d'association

Dolibarr est certes installé, mais pas encore opérationnel; je bute sur l'export des données. Je m'y consacrerai dès que possible.

## Les Cahiers

Jean-Michel Hufflen, le nouveau rédacteur en chef des Cahiers, s'est entouré d'un comité de rédaction renouvelé. Le prochain numéro des Cahiers sort en septembre et portera le numéro 58. Il sera adressé aux adhérents à jour de cotisation ainsi qu'à ceux qui s'étaient abonnés depuis 2012.

La parution de ce nouveau numéro nous a fait décider d'ouvrir à la publicité les colonnes de la présente publication, comme on le voit en page 37.

Patrick BIDEAULT

---

32. La publication de ces lignes m'a fait réserver l'un de ces noms de domaine. Dans l'éventualité où il serait choisi, je le rétrocéderai à l'association – à prix coûtant, cela va sans dire.

33. L'indépendance des échanges de la liste GUT n'empêche pas l'association d'y faire quelques annonces, comme celle de la sortie de la présente Lettre.

## 🌀 QUELQUES NOUVELLES DE POLOGNE

Suite à notre article de la Lettre précédente, consacré à la fonte Schola, qui servit à la composer, nous recevons des nouvelles du GUST, le *Polska Grupa Użytkowników Systemu TeX*, ou groupe polonophone des utilisateurs de TeX. Les homologues polonais de notre association nous donnent, par la voix de Jerzy Ludwichowski, les dernières nouvelles concernant leur e-fonderie.

Ainsi donc, la fonderie du GUST n’a pas chômé :

- les nouvelles versions de TeX Gyre Adventor et Pagella sont disponibles ;
- les fontes Bonum, Pagella et Schola disposent désormais d’un support mathématique complet ;
- il en va de même pour Latin Modern, dont un bug a été corrigé récemment (cette correction est intégrée à la distribution TeX Live 2021).

Les nouvelles versions de Bonum, Chorus, Cursor, Heros, Schola et Termes sont en cours de fabrication.

La fonderie du GUST a bénéficié du concours financier de quelques groupes d’utilisateurs et de particuliers. Les fonds récoltés permettent de rémunérer les dessinateurs de caractères, qui sont des typographes indépendants travaillant selon les directives du GUST, qui leur fournit un support technique si nécessaire.

Il est possible de contribuer à la création de ces fontes (rappelons que le GUST est à l’origine de la mise à disposition de *milliers* de glyphes) :

- soit en adressant un don au TUG, en spécifiant bien que ces fonds sont à destination de la fonderie du GUST ;
- soit, tout simplement, en effectuant un virement SEPA en euros vers le compte du GUST.

Titulaire :	Polska Grupa Uzytkownikow Systemu TeX – GUST
IBAN :	PL91 1160 2202 0000 0000 5530 4977
Référence du virement :	GUST e-foundry fonts contribution

Algotype, le programme utilisé pour créer les caractères de la fonderie, est composé de trois parties :

- Algotype MetaPost, qui est utilisé pour la création initiale des glyphes ;
- Algotype Fontforge, qui les finalise ;
- et Algotype Python, qui fait travailler ensemble les deux logiciels :
  - il contrôle le fonctionnement de MetaPost ;
  - il convertit les données générées en données utilisables par FontForge ;
  - il fait alors appel à FontForge...
  - dont il corrige, si besoin est, les fichiers que ce logiciel produit !

Les auteurs d’Algotype, Bogusław Jackowski et Marek Ryćko, sont conscients de la nécessité de rédiger de bonnes documentations pour leurs logiciels, de manière à ce qu’ils puissent servir à d’autres utilisateurs. Ces documentations verront le jour peu à peu.

Le GUST crée donc des biens librement utilisables par tous. Cet apport est incommensurable, tant par le nombre de glyphes mis à la disposition des utilisateurs que par les services que de telles fontes leur rendent. Notons ici que cet apport désintéressé est le fait d’un groupe d’utilisateurs, travaillant ensemble pour le bien commun. C’est un exemple, s’il en était besoin, de l’utilité du travail associatif.

## APERÇU DU PACKAGE NICEMATRIX

### L'origine : les lignes pointillées

Dans ses premières versions (février 2018), l'extension `nicematrix`<sup>34</sup> se proposait d'améliorer l'esthétique des matrices comportant des points de suspension. De là vient le nom `nicematrix` alors que le nom `nicetabular` serait plus adapté pour les versions actuelles.

Traditionnellement, quand on utilise l'`amsmath`, voici comment on compose la matrice triangulaire supérieure ne comportant que des 1 au-dessus de la diagonale :

code	résultat
<pre> 1  \$\begin{pmatrix} 2  1      &amp; \cdots &amp; \cdots &amp; 1      \\ 3  0      &amp; \ddots &amp;      &amp; \vdots \\ 4  \vdots &amp; \ddots &amp; \ddots &amp; \vdots \\ 5  0      &amp; \cdots &amp; 0      &amp; 1      \\ 6  \end{pmatrix}\$ </pre>	

Les lignes en pointillées sont discontinues, ce qui n'est pas étonnant vu que l'on n'a placé que des caractères isolés comme on le ferait en typographie au plomb. Pour arriver à avoir des lignes pointillées continues, on est obligé d'utiliser une couche de programmation graphique capable de dessiner des objets au-dessus des caractères. L'extension `nicematrix` utilise pour cela `PGF`<sup>35</sup>, qui est une sous-couche de `TikZ`. Avec l'environnement `pNiceMatrix` de `nicematrix` et les commandes `\Cdots`, `\Vdots` et `\Ddots` qui y sont définies, on obtient le résultat suivant.

code	résultat
<pre> 1  \$\begin{pNiceMatrix} 2  1      &amp; \Cdots &amp; \Cdots &amp; 1      \\ 3  0      &amp; \Ddots &amp;      &amp; \Vdots \\ 4  \Vdots &amp; \Ddots &amp; \Ddots &amp; \Vdots \\ 5  0      &amp; \Cdots &amp; 0      &amp; 1      \\ 6  \end{pNiceMatrix}\$ </pre>	

Comme on le voit, la syntaxe est très proche de celle de l'environnement `pmatrix` de l'`amsmath`.

En fait, certaines commandes sont redondantes (il n'y a pas besoin de deux instructions `\Cdots` pour tracer une même ligne pointillée horizontale : une seule suffit).

code	résultat
<pre> 1  \$\begin{pNiceMatrix} 2  1      &amp;      &amp; \Cdots &amp; 1      \\ 3  0      &amp; \Ddots &amp;      &amp; \Vdots \\ 4  \Vdots &amp; \Ddots &amp;      &amp;      \\ 5  0      &amp; \Cdots &amp; 0      &amp; 1      \\ 6  \end{pNiceMatrix}\$ </pre>	

Pour arriver à ce résultat, l'extension `nicematrix` a créé un nœud<sup>36</sup> `PGF-TikZ` derrière le contenu de chaque cellule de la matrice, *exception faite des cellules dont le contenu est vide*.

34. L'extension `nicematrix` est écrite au maximum en `expl3`, qui est, en quelque sorte, le langage de programmation de `LaTeX3` et qui a, d'ores et déjà, été intégré au noyau `LaTeX`.

35. *Portable Graphics Format* [anglais] : format de graphiques portables.

36. Avec `PGF-TikZ`, les nœuds sont créés, par exemple, avec l'opération `node` dans un chemin.

Sur la figure suivante, on a colorié en gris ces nœuds PGF-TikZ.

$$\begin{pmatrix} \mathbf{1} & \dots & \dots & \mathbf{1} \\ \mathbf{0} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \mathbf{0} & \dots & \mathbf{0} & \mathbf{1} \end{pmatrix}$$

Ces nœuds sont utilisés en interne par `nicematrix` pour tracer les lignes pointillées continues (entre autres choses), mais l'utilisateur final peut aussi les utiliser directement avec TikZ. Dans le « `\CodeAfter` » de l'environnement, le nœud correspondant au contenu de la cellule en ligne  $i$  et colonne  $j$  est simplement noté  $i - j$ .

code	résultat
<pre> 1  \$\begin{pNiceMatrix} 2  1 &amp; 2 &amp; 3 \\ 3  4 &amp; 5 &amp; 6 \\ 4  7 &amp; 8 &amp; 9 \\ 5  \CodeAfter 6  \tikz \draw (2-2) circle (2mm) ; 7  \end{pNiceMatrix}\$ </pre>	$\begin{pmatrix} 1 & 2 & 3 \\ 4 & \textcircled{5} & 6 \\ 7 & 8 & 9 \end{pmatrix}$

En créant ces nœuds, `nicematrix` est assez proche de ce que propose la bibliothèque `matrix` de TikZ. Il faut néanmoins remarquer que l'espacement entre les rangées d'une matrice de TikZ n'obéit pas aux mêmes règles que celles qui sont à l'œuvre dans l'environnement `pNiceMatrix` (qui est composé exactement comme l'environnement `pmatrix`, une `pNiceMatrix` étant, en fait, une `pmatrix` en interne).

Voici, à titre d'exemple, une matrice composée avec un environnement `NiceMatrix` de `nicematrix` puis avec une `matrix` de TikZ.

code	résultat
<pre> 1  \$\begin{NiceMatrix} 2  \frac{12}{2} &amp; 2 \\ 3  3 &amp; 4 \\ 4  \end{NiceMatrix}\$ </pre>	$\begin{matrix} \frac{1}{2} & 2 \\ 3 & 4 \end{matrix}$

code	résultat
<pre> 1  \$\vcenter{ 2  \begin{tikzpicture} 3  \matrix (A) [matrix of math nodes] 4  { 5  \frac{12}{2} &amp; 2 \\ 6  3 &amp; 4 \\ 7  } ; 8  \end{tikzpicture}}\$ </pre>	$\begin{matrix} \frac{1}{2} & 2 \\ 3 & 4 \end{matrix}$

Même si, à l'origine, l'extension `nicematrix` avait pour but de composer des matrices mathématiques, elle permet aussi, dans les versions récentes, de composer des tableaux les plus généraux (non mathématiques), comportant, en particulier, des filets.

## Les nœuds correspondants à la position des filets

Ayant créé des nœuds pour le contenu des cellules, comme dit précédemment, l'auteur de `nicematrix` (c'est-à-dire moi-même) a eu l'idée de créer des nœuds de type `coordinate` correspondants à la position des filets. Le nœud se trouvant en position  $i$  sur la diagonale

est nommé simplement `i` (PGF autorise pour les nœuds un tel nom composé uniquement de chiffres).

	tulipe	lys
arum		violette mauve
muguet	dahlia	

En fait, ce nœud n’est pas tout à fait de type `coordinate`, comme on vient de le dire, car il a une ancre nommée 5 (PGF autorise un tel nom ne comportant que des chiffres pour une ancre). L’ancre 5 du nœud `i`, qui se note `i.5`, se trouve à mi-chemin entre le nœud `i` et le nœud `i + 1`. On peut donc considérer la figure suivante et imaginer, grâce à cette astuce, qu’il existe des nœuds `1.5`, `2.5` et `3.5`.

	tulipe	lys
arum		violette mauve
muguet	dahlia	

Pour avoir l’intersection du filet horizontal numéro `i` et du filet vertical numéro `j`, il suffit d’utiliser le raccourci syntaxique « `-|` » de TikZ : « `i-|j` » est le point en ligne `i` et colonne `j`. On donne quelques exemples ci-dessous.

	tulipe	lys
arum		violette mauve
muguet	dahlia	

Bien sûr, il ne faut pas confondre une écriture `i-j` qui désigne un nœud rectangulaire (avec ses ancres `north`, `south`, `west`, `east`, etc.) et une écriture `i-|j` qui désigne un point géométrique.

L’utilisation de ces nœuds autorise bien entendu une gamme de possibilités graphiques qui seraient très difficiles, voire impossibles, à obtenir avec les tableaux classiques de LaTeX. En ce qui concerne l’utilisateur final, il peut utiliser ces nœuds avec TikZ dans le « `\CodeAfter` » des environnements de `nicematrix`. L’exemple suivant montre comment barrer en croix une case d’un tableau `NiceTabular` (qui est le pendant dans `nicematrix` de l’environnement classique `tabular`).

code

```

1 \begin{NiceTabular}{ccc}[hvlines]
2 Laurence & Stéphanie & Noémie \\
3 Alexandra & & Charlotte \\
4 Cécile & Eugénie & Emmanuelle \\
5 \CodeAfter
6 \tikz
7 \draw (2-|2) -- (3-|3) (3-|2) -- (2-|3) ;
8 \end{NiceTabular}

```

résultat

Laurence	Stéphanie	Noémie
Alexandra	X	Charlotte
Cécile	Eugénie	Emmanuelle

La clé `hvlines` trace tous les filets horizontaux et verticaux.

L’extension `nicematrix` utilise en interne ces nœuds pour tracer les filets. Cela permet, par exemple, d’avoir des filets verticaux qui traversent les double-filets horizontaux tracés par `\hline\hline`, sans qu’il y ait besoin d’utiliser l’extension `hhline`.

code	résultat						
<pre> 1 \begin{NiceTabular}{ c c } \hline 2 Premier &amp; Deuxième \\ \hline\hline 3 Paul &amp; \\ \hline 4 Marie &amp; Pauline \\ \hline 5 \end{NiceTabular} </pre>	<table border="1"> <tr> <td>Premier</td> <td>Deuxième</td> </tr> <tr> <td>Paul</td> <td></td> </tr> <tr> <td>Marie</td> <td>Pauline</td> </tr> </table>	Premier	Deuxième	Paul		Marie	Pauline
Premier	Deuxième						
Paul							
Marie	Pauline						

Bien entendu, l’auteur est conscient que, dans bien des situations, une présentation de tableau avec uniquement des filets horizontaux telle que proposée par l’extension `booktabs` de Simon FEAR est préférable. Nous présentons ici des outils sans préjuger des choix graphiques des utilisateurs.

## Le problème du coloriage des tableaux

Fort de tous ces nœuds, `nicematrix` peut essayer de faciliter la composition des tableaux.

L’un des problèmes subtils de la composition des tableaux est celui du coloriage. L’outil le plus classique utilisé pour cela est l’extension `colortbl` de David CARLISLE. Beaucoup de personnes chargent cette extension *via* l’extension `xcolor` avec son option `table`.

L’extension `colortbl` suit la construction des tableaux faite par  $\LaTeX$ . Le tableau est donc construit ligne par ligne, les panneaux de couleur étant insérés au moment adéquat, entre les filets et les composantes des cellules. Pour une ligne colorée qui suit un filet horizontal, les panneaux de couleur sont insérés après le filet horizontal. La taille et la position de ces objets sont calculés, comme toujours avec  $\TeX$ , à la précision du point d’échelle (sp) et le résultat, dans le PDF produit, est mathématiquement irréprochable.

Néanmoins, les lecteurs de PDF qui ont à effectuer la rasterisation des pages (c’est-à-dire la conversion des contenus vectoriels en images pixellisées pour l’affichage sur écran) ont tendance à raisonner selon le « modèle du peintre » (*painting model* en anglais) qui n’est pas celui de la typographique.

En typographie, les différents types (c’est-à-dire les petits blocs de plomb correspondants aux caractères) sont juxtaposés les uns à côté des autres sans pouvoir se chevaucher.  $\TeX$  procède essentiellement de la même manière pour la construction de ses « boîtes ».

En revanche, le PostScript (dont le PDF est l’héritier) décrit une page selon le « modèle du peintre ». La page est constituée de « marques » qui se superposent opaquement comme le font les coups de pinceau d’un peintre. Cela permet, bien entendu, la réalisation aisée de figures et d’illustrations.

C’est pourquoi certains lecteurs de PDF ont tendance à donner la priorité aux éléments tracés postérieurement aux autres. Un rectangle coloré tracé après un filet horizontal, les deux étant mathématiquement juxtaposés sans aucun recouvrement, aura quand même tendance à masquer le filet à certains niveaux de zoom. C’est le cas en particulier avec Adobe Reader.

À l’heure où de nombreux PDF ne sont jamais imprimés mais uniquement visualisés sur écran, on peut légitimement vouloir résoudre ce problème. L’extension `nicematrix` apporte une solution car, chose que nous n’avions pas encore mentionnée, les nœuds de la forme « `i-j` » sont en fait accessibles *avant* la construction du tableau (ce n’est pas encore le cas pour les nœuds de la forme « `i-j` »)<sup>37</sup>. Pour y parvenir, `nicematrix` utilise un mécanisme proposé par PGF (c’est le même mécanisme que celui utilisé par l’extension `tikzmark`) : les coordonnées des nœuds sont écrites dans le fichier `.aux` pour pouvoir être accessibles, à la

37. En fait, dans les dernières versions de `nicematrix`, ces nœuds-là sont également accessibles (quand on a utilisé la clé `create-cell-nodes` de la commande `\CodeBefore`).

compilation suivante, avant le début de l’environnement (le fichier aux est toujours, avec  $\LaTeX$ , exécuté au début de la compilation).

L’utilisateur final peut d’ailleurs, s’il le souhaite, utiliser ces nœuds avant la construction du tableau proprement dit. Une syntaxe spéciale est prévue pour cela par `nicematrix`, utilisant les mots-clés `\CodeBefore` et `\Body`. On en donne maintenant un exemple.

code

```

1 \begin{NiceTabular}{ccc}[hvlines,rules/width=0.2pt,baseline=1]
2   \CodeBefore
3     \tikz
4     \fill [red!15] (2-|2.5) -- (2.5-|2) -- (3-|2.5) -- (2.5-|3) -- cycle ;
5   \Body
6     dahlia & narcisse & primevère \\
7     tulipe & rose & muguet \\
8     camélia & iris & violette
9 \end{NiceTabular}

```

résultat

dahlia	narcisse	primevère
tulipe	rose	muguet
camélia	iris	violette

L’extension `nicematrix` utilise en interne ce `\CodeBefore` (si on peut dire) pour colorier les cellules, les rangées et les colonnes avec une syntaxe similaire à celle de `colortbl` (à condition d’utiliser la clé `colortbl-like`).

code

```

1 \NewDocumentCommand { \Blue } { } {\columncolor{blue!15}} % syntaxe de xparse
2 \begin{NiceTabular}[colortbl-like]{>{\Blue}c>{\Blue}cc}
3   \toprule \rowcolor{red!15}
4   Nom & Prénom & Année de naissance \\
5   \midrule
6   Achard & Jacques & 5 juin 1962 \\
7   Lefebvre & Mathilde & 23 mai 1988 \\
8   Vanesse & Stéphanie & 30 octobre 1994 \\
9   Dupont & Chantal & 15 janvier 1998 \\
10  \bottomrule
11 \end{NiceTabular}

```

résultat

Nom	Prénom	Année de naissance
Achard	Jacques	5 juin 1962
Lefebvre	Mathilde	23 mai 1988
Vanesse	Stéphanie	30 octobre 1994
Dupont	Chantal	15 janvier 1998

Les commandes `\toprule`, `\midrule` et `\bottomrule` sont des commandes de l’extension `booktabs`. On remarquera en passant que les commandes de coloriage de `nicematrix` sont compatibles avec ces commandes de `booktabs`.

## Formatage à l’intérieur ou à l’extérieur du tableau

Il y a toujours, pour les langages de description de tableaux, l’alternative entre insérer les instructions de formatage au fur et à mesure des composantes du tableau ou bien les avoir de manière centralisée, avant ou après les données proprement dites.



Sur bien des points, l’extension `nicematrix` offre à l’utilisateur les deux possibilités.

Par exemple, concernant les coloriages de rangées, de colonnes ou de cellules, on a vu que des commandes à la syntaxe similaire à celles de `colortbl` sont proposées (sans compter qu’il est possible d’utiliser tout simplement les commandes de `colortbl` elles-mêmes dans un tableau `NiceTabular`). Pour autant, des commandes de coloriage sont également disponibles dans l’argument de la commande `\CodeBefore` pour spécifier le coloriage de manière centralisée.

Le résultat est bien entendu le même que précédemment.

code

```

1 \begin{NiceTabular}{ccc}
2 \CodeBefore
3 \columncolor{blue!15}{1,2}
4 \rowcolor{red!15}{1}
5 \Body
6 \toprule
7 Nom & Prénom & Année de naissance \\
8 \midrule
9 Achard & Jacques & 5 juin 1962 \\
10 Lefebvre & Mathilde & 23 mai 1988 \\
11 Vanesse & Stéphanie & 30 octobre 1994 \\
12 Dupont & Chantal & 15 janvier 1998 \\
13 \bottomrule
14 \end{NiceTabular}

```

Nom	Prénom	Année de naissance
Achard	Jacques	5 juin 1962
Lefebvre	Mathilde	23 mai 1988
Vanesse	Stéphanie	30 octobre 1994
Dupont	Chantal	15 janvier 1998

résultat

Concernant les filets, il est possible de les spécifier de la manière classique (avec le spécificateur « | » dans le préambule et la commande `\hline`). Il existe néanmoins aussi une clé `hvlines` qui trace tous les filets. Sans compter qu’on peut toujours tracer n’importe quel trait avec TikZ en utilisant les nœuds créés par `nicematrix`.

## La commande `\Block`

La commande `\Block` est l’outil le plus versatile disponible dans les environnements de `nicematrix`. Cette commande s’utilise dans une cellule du tableau avec la syntaxe suivante où les deux premiers arguments, entre crochets et chevrons, sont optionnels :

```
\Block[⟨options⟩]⟨tokens⟩{⟨n⟩-⟨p⟩}{⟨contenu⟩}
```

Les nombres  $n$  et  $p$  désignent les nombres de lignes et de colonnes du bloc à créer. Si cet argument est laissé vide, c’est un bloc de taille 1-1 qui est créé.

La commande `\Block` sert en premier lieu à fusionner un certain nombre de cellules du tableau en plaçant son contenu au centre du rectangle de cases fusionnées. De ce fait, elle peut avantageusement remplacer les commandes classiques `\multicolumn` et `\multirow` (cette dernière étant proposée par l’extension éponyme `multirow`).

code

```

1  $\begin{bNiceArray}{ccc|c}[margin]
2    \Block{3-3}<\LARGE>{A} & & & 0 & \\
3    & \hspace*{1cm} & & \vdots & \\
4    & & & 0 & \\ \hline
5    0 & \cdots & & 0 & 0
6  \end{bNiceArray}$

```

$$\left[ \begin{array}{ccc|c} & A & & 0 \\ & & & \vdots \\ & & & 0 \\ \hline 0 & \dots & 0 & 0 \end{array} \right]$$

résultat

Les tokens placés entre chevrons sont insérés avant le passage en mode mathématique (ici, la commande `\LARGE` n'aurait pas pu être utilisée en mode mathématique).

Le contenu du bloc est placé *après* la composition du tableau en calculant sa position grâce aux nœuds PGF-TikZ créés par `nicematrix`. De la sorte, le placement est correct dès lors que l'on a bien spécifié le nombre de rangées logiques du tableau (et le nombre de colonnes, bien entendu).

Cela est à comparer avec le fonctionnement de la commande `\multirow` qui place son contenu au moment de composer la ligne où apparaît la commande `\multirow` et qui a donc besoin en argument du nombre de *lignes physiques* à prendre en compte.

Voici d'ailleurs ce que l'on écrirait sans doute pour composer la même matrice avec `\multirow`. On a été obligé d'ajuster « à la main » le premier argument de `\multirow` (3.5).

code

```

1  $\left[
2    \begin{array}{ccc|c}
3      \multicolumn{3}{c}{\multirow{3.5}{*}{\LARGE $A$}} & 0 \\
4      & \hspace*{1cm} & & \vdots \\
5      & & & 0 \\ \hline
6      0 & \cdots & & 0 & 0
7    \end{array}
8  \right]$

```

$$\left[ \begin{array}{ccc|c} & A & & 0 \\ & & & \vdots \\ & & & 0 \\ \hline 0 & \dots & 0 & 0 \end{array} \right]$$

résultat

À l'origine, la commande `\Block` avait seulement pour but de composer des matrices par blocs comme dans l'exemple précédent. Mais ses usages se sont bien étendus au fur et à mesure des versions successives de `nicematrix`. On peut même avoir à l'utiliser avec un contenu vide. En effet, les positions des blocs sont gardées en mémoire par `nicematrix` et influencent le comportement de plusieurs autres commandes. Par exemple, la clé `hvlines` dont nous avons déjà parlé trace en fait tous les filets, mais exception faite de ceux à l'intérieur des blocs. En un sens, on peut dire qu'elle « respecte » les blocs.

code	résultat
<pre> 1 \begin{NiceTabular}{cccc}[hvlines] 2 &amp; &amp; &amp; \\ 3 &amp; \Block{2-2}{} \\ 4 &amp; &amp; &amp; \\ 5 &amp; &amp; &amp; \\ 6 \end{NiceTabular} </pre>	

On peut par exemple aussi utiliser la commande `\Block` pour mettre en évidence une partie d'une matrice.

code	résultat
<pre> 1 \$\begin{pNiceMatrix}[right-margin] 2 1 &amp; 2 &amp; 3 \\ 3 4 &amp; \Block[draw,rounded-corners]{2-2}{} \\ 4 0 &amp; 0 \\ 5 7 &amp; 0 &amp; 0 \\ 6 \end{pNiceMatrix}\$ </pre>	

## L'avenir de `nicematrix`

Bien qu'elle facilite la composition des tableaux et des matrices, l'extension `nicematrix` n'est pas l'extension universelle de construction de tableaux dont rêvent sans doute les utilisateurs de  $\text{\LaTeX}$ .

Pour le moment (version 5.17a) les principaux défauts de l'extension `nicematrix` sont les suivants :

- il n'est pas possible d'utiliser un environnement de `nicematrix` à l'intérieur d'un autre environnement de `nicematrix` (mais on peut utiliser un environnement de `nicematrix` à l'intérieur d'un `tabular` et vice versa, ce qui fait que ce point n'est sans doute pas très gênant);
- les tableaux construits par `nicematrix` ne peuvent pas être coupés par des sauts de page.
- il n'y a pas d'environnement « `NiceTabularX` », c'est-à-dire qu'il n'y a pas de système de calcul automatique de largeur de colonne comme celui proposé par l'extension `tabularx` de David CARLISLE avec ses colonnes de type `X`.

Ce dernier point sera sans doute corrigé dans une nouvelle version de `nicematrix` qui proposera un type de colonne `X`. Arriver à proposer des environnements sécables par saut de page (comme le propose par exemple l'extension `longtable`) est en revanche un objectif nettement plus difficile à atteindre.

# CAHIERS GUTenberg

Les Cahiers GUTenberg reparaissent enfin !

Au sommaire du prochain numéro :

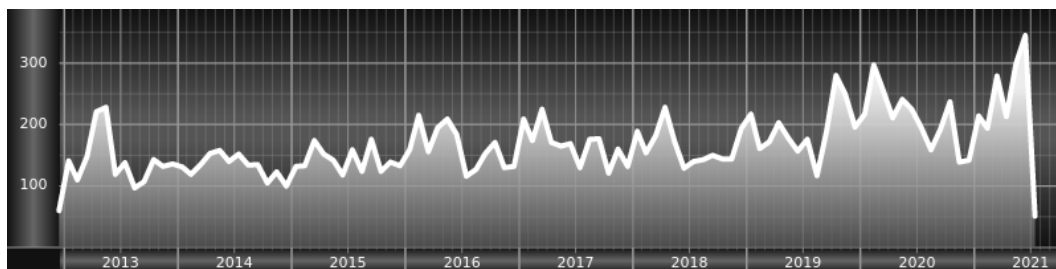
- *Donald Knuth : des mathématiques à la typographie*, par Sandrine Chenevez, Myriam Hamla & Stéphanie Louison
- *Imprimer ou ne pas imprimer*, par Peter Flynn
- *LaTeX au collège? Une avancée*, par Christophe Poulain
- *LuaTeX et Metapost avec Luamplib, une introduction*, par Maxime Chupin <sup>a</sup>
- *Histoire des altérations musicales*, par Jean-Michel Hufflen <sup>a</sup>

Ce numéro sera publié en septembre 2021. Pour célébrer le renouveau des Cahiers, ce numéro sera exceptionnellement adressé à tous les adhérents à jour de cotisation ainsi qu'à ceux qui s'étaient abonnés depuis 2012.

Il est aussi possible de le commander en écrivant au secrétariat de l'association, avec vos nom, adresse et nombre d'exemplaires souhaités.

<sup>a</sup>. Maxime et Jean-Michel sont administrateurs de l'association.

## 🎵 ET MAINTENANT, UNE BONNE VIEILLE VEILLE TECHNOLOGIQUE!



*Chers adhérents, nous veillons technologiquement pour vous!*

*En effet, la présente rubrique est dédiée aux nouveautés apparues sur le CTAN<sup>38</sup> que vous auriez pu manquer. Elle témoigne de la vitalité de la communauté T<sub>E</sub>X. Nous y listerons la grande majorité des packages ou classes récemment apparus ainsi que parfois, parmi ceux « simplement » mis à jour, certains qui méritent à notre sens d'être signalés. Nous ne nous interdirons pas, le cas échéant, d'en mentionner de plus anciens, soit parce qu'ils nous semblent injustement méconnus, soit parce qu'ils sont les fruits de contributeurs francophones. Au sujet de la francophonie, nous signalons au moyen du logo 🌐 les travaux de francophones.*

### Nouveautés


Pour la rubrique du présent numéro, nous listons la plupart des nouveautés, classées par ordre (quasi) chronologique, apparues depuis la précédente *Lettre* (numérotée 43 et parue le 10 avril 2021).

**beamerthemelalic** : est un thème beamer destiné aux documents du *Laboratório de Linguística e Inteligência Computacional (Lalic)* de l'université fédérale de São Carlos, dans l'état de São Paulo, au Brésil.

**ninecolors** : fournit des ensembles de neuf couleurs pour 13 teintes (*hue* en anglais) chacune. Toutes les couleurs avec le même suffixe (*gray8*, *blue8*, etc.) ont même un niveau de luminance. Cela permet de sélectionner des couleurs avec un contraste WCAG<sup>39</sup> correct.

**xintsession** : permet d'ouvrir des sessions de calcul interactives avec *etex* (ou *pdftex*) exécuté en ligne de commande, basées sur les packages *xintexpr* et *polexpr*.

**mahjong** : fournit une interface permettant la composition de *tuiles* de mah-jong. Le package utilise la notation *MPSZ* et offre de nombreuses possibilités de composition.

<i>code</i>
<code>\mahjong{77"7z}</code>

<i>résultat</i>

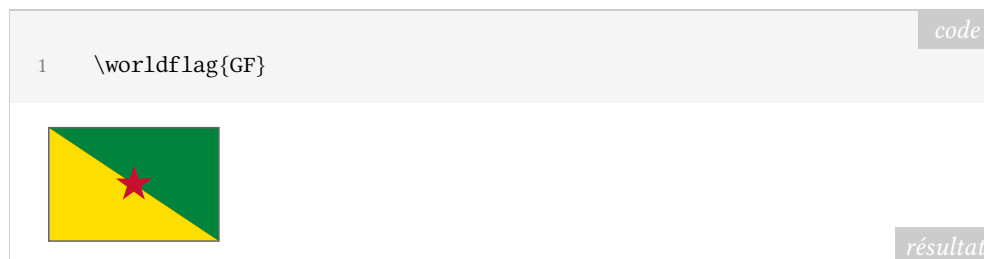
**huawei** : est une classe permettant de composer des documents techniques de travail et des présentations pour celles et ceux qui travaillent avec et pour Huawei. Logo, fonte, couleurs, le tout est pré-configuré pour s'accorder aux standards de l'entreprise.

**byo-twemojis** : est un outil de fabrication d'émoticônes personnalisées.

38. *Comprehensive T<sub>E</sub>X Archive Network* [anglais] : réseau complet d'archives T<sub>E</sub>X.

39. *Web Content Accessibility Guidelines* [anglais] : directives sur l'accessibilité du contenu Web.

**worldflags** : fournit les commandes pour dessiner les drapeaux avec TikZ. Très complet, ce package permet de composer très simplement les drapeaux nationaux du monde entier ; la plupart d’entre eux sont immédiatement disponibles, et accessibles selon leur domaine internet, comme en témoigne ci-dessous le code et le drapeau y afférant, que chacun reconnaîtra :



**tlmgr-intro-zh-cn** : est une traduction en chinois du document *Basic Usage of tlmgr, the T<sub>E</sub>X Live Manager*. Ce document présente certaines des fonctionnalités les plus courantes de tlmgr.

**tipauni** : redéfinit la plupart des commandes TIPA<sup>40</sup> du package **tipa** (utilisant l’encodage T3) pour utiliser les caractères unicodes avec Lua<sub>L</sub>T<sub>E</sub>X et X<sub>L</sub>L<sub>A</sub>T<sub>E</sub>X. Ces commandes permettent de composer les symboles de plusieurs systèmes de transcription phonétique.

**fontsetup-nonfree** : est la partie non libre du package **fontsetup-nonfree** qui est une surcouche au package **fontspec** qui rend la configuration des fontes facile et rapide pour X<sub>L</sub>L<sub>A</sub>T<sub>E</sub>X et Lua<sub>L</sub>T<sub>E</sub>X. Il suffit de charger le package en utilisant l’une des fontes supportées comme option. Ici, cela concerne des fontes qui ne sont pas libres.

**coffeestains** : nul ne sait à quoi sert ce package.

**codehigh** : permet de composer les codes sources avec coloration syntaxique. Ce package utilise le package de L<sub>A</sub>T<sub>E</sub>X3 **l3regex**, même si avec Lua<sub>L</sub>T<sub>E</sub>X, il permet aussi d’utiliser la bibliothèque Lua LPEG (*Parsing Expression Grammars for Lua*). L’auteur prétend que son package est plus puissant que **listings** et plus simple d’utilisation que **minted**, cependant le package est encore en phase de développement.

**projlib** : regroupe diverses fonctionnalités utilisées par les autres packages de ce prolifique auteur : **minimalist**, **einfart**, **simplivre**, **colorist**, **lebhart** et **beaulivre** ; ils offrent tous un support natif de l’anglais, du français, de l’allemand, du chinois et d’autres langues. ProjLib donne accès à des commandes facilitant la composition des dates, des références de auteurs, des théorèmes etc. Il est à noter que ce package dispose d’une documentation en français<sup>41</sup>.

**tabularray** : est un package de composition de tableau, écrit en L<sub>A</sub>T<sub>E</sub>X3. Ce package est remarquable et est sans doute une révolution dans les outils pour les tableaux en L<sub>A</sub>T<sub>E</sub>X. En effet, au lieu de modifier les commandes de table internes de L<sub>A</sub>T<sub>E</sub>X, comme les font classiquement les autres packages du domaine, l’auteur a abandonné l’utilisation des commandes T<sub>E</sub>X de bases telles que **\halign**, **\noalign**, etc. et a utilisé les fonctions L<sub>A</sub>T<sub>E</sub>X3, langage désormais mature, pour analyser le tableau puis le composer. En gardant une compatibilité avec la syntaxe de base des tableaux L<sub>A</sub>T<sub>E</sub>X 2<sub>ε</sub>, ce package sépare complètement le contenu du style du tableau, ancrant davantage L<sub>A</sub>T<sub>E</sub>X dans la séparation fond/forme. De plus, le style du tableau peut être entièrement défini par le système classique, mais tellement pratique, de clé/valeur. L’exemple suivant illustre la puissance de ce package.

40. T<sub>E</sub>X IPA or Tokyo IPA [anglais] : alphabet phonétique international.

41. La rédaction remercie Xu Jinwen, l’auteur de ce package, qui a gentiment reformulé notre description de son travail.

code

```

1 % couleurs utilisées : fournies par le package `ninecolors'
2 \begin{tblr}{width=0.8\textwidth,
3   colspec={1 X[2] X[3] X[-1]}, % spécification des colonnes
4   row{odd} = {bg=azure8},      % lignes impaires
5   % première ligne
6   row{1} = {bg=azure3, fg=white, font=\sffamily,c},
7   % première colonne
8   column{1} = {bg=purple7},
9   % filets horizontaux
10  hlines = {1pt,white},
11  }
12 Alpha & Beta & Gamma & Delta \\
13 Epsilon & Zeta & Eta & Theta \\
14 Iota & Kappa & Lambda & Mu \\
15 Nu & Xi & Omicron & Pi \\
16 Rho & Sigma & Tau & Upsilon \\
17 Phi & Chi & Psi & Omega \\
18 \end{tblr}

```

Alpha	Beta	Gamma	Delta
Epsilon	Zeta	Eta	Theta
Iota	Kappa	Lambda	Mu
Nu	Xi	Omicron	Pi
Rho	Sigma	Tau	Upsilon
Phi	Chi	Psi	Omega

résultat

**pdfextra** : est un package additionnel pour le format OpTeX permettant d'avoir accès aux fonctionnalités plus avancées de PDF, en particulier concernant l'interactivité et les fonctionnalités multimedia.

**tiscreen** : permet de reproduire l'écran des célèbres calculettes TI-82.

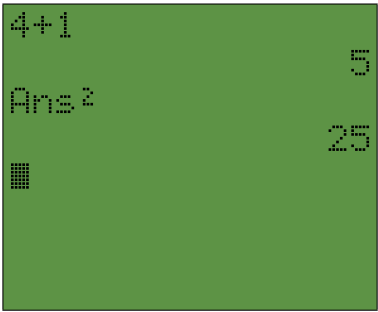
code

```

1
2
3 \tiscreen
4 |4+1      |
5 |         |5|
6 |Ans{sq}  |
7 |         |25|
8 |{fcur}   |
9 |         |
10 |         |
11 |         |

```

résultat



**pgfmath-xfp** : permet de définir des fonctions pgfmath en les évaluant avec le package xfp.

**bmstu-iu8** : est une classe permettant de composer des rapports pour le département « sécurité informatique » (IU-8) de l'université technique d'État de Moscou-Bauman (BMSTU).

**bjfuthesis** : est une classe pour composer les thèses des étudiants et étudiantes de l'université Beijing Forestry (BJFU). La classe est basée sur la classe ctexbook.



- biblatex-cv** : permet de créer un *curriculum vitae* (CV) à partir d'un fichier `.bib`. Le package utilise le couple `biblatex/biber` pour formater, grouper et trier les entrées du CV.
- kdpcover** : aide à produire des couvertures de livre pour la plateforme *Kindle Direct Publishing* (KDP), l'ajustement de la largeur de la tranche étant fait automatiquement.
- bangla** : fournit des commandes permettant de composer du texte dans la langue bengalie.
- wichura-table** : est une collection de macros  $\TeX$  (et non  $\LaTeX$ ) qui facilitent la composition de tableaux comme les tableaux de transferts budgétaires.
- minim** : est un format moderne du type de Plain  $\TeX$  pour le moteur  $\text{Lua}\TeX$  comprenant un support bas niveau pour de nombreuses extensions de  $\text{Lua}\TeX$  et des nouvelles possibilités de PDF. Ce format peut-être utilisé comme un remplaçant de Plain  $\TeX$ , mais il semble plus utile comme une base pour d'autres formats.
- minim-mp** : intègre la gestion de la bibliothèque `Lua mp1ib` au format `minim`. Il permet aussi d'utiliser `luatex` directement comme d'un processeur `METAPOST`.
- minim-math** : configure le format `minim` pour l'utilisation des fontes mathématiques OpenType et l'utilisation du codage Unicode.
- minim-pdf** : ajoute le support bas niveau au format `minim` pour la gestion de la structure d'un document PDF, permettant ainsi de rendre les fichiers PDF produits conforme au format PDF/A.
- minim-hatching** : est une petite bibliothèque `METAPOST` pour la création de pavage monochrome avec le processeur `METAPOST` `minim-mp`.
- minim-xmp** : permet l'inclusion simple de paquets XMP (*eXtensible Metadata Platform*) et génère automatiquement les schémas de l'extension PDF/a.
- ffcode** : fournit des commandes et environnements pour composer du code source pour les articles académiques de façon sobre et élégante. Ce package utilise les packages `minted` et `tcolorbox`.
- href-ul** : permet de composer les liens hypertextes avec la commande `\href` en soulignés de façons à ce qu'ils ressemblent à ce qu'on peut voir sur le Web.
- arabic-book** : est une classe permettant de composer un document de type book en arabe. Cette classe est pour le moteur  $\text{Xe}\TeX$ .
- apalike-ejor** : fournit le style bibliographique  $\text{BIB}\TeX$  `apalike-ejor.bst` pour suivre les conventions pour le *European Journal of Operational Research*. Il s'agit d'un fork du package `apalike`.
- acrotex-js** : rassemble des fichiers JavaScript utilisés par le package `acrotex`.
- docassembly** : donne accès à quelques fonctionnalités de l'API du JavaScript d'Acrobat concernant des aspects de sécurité.
- optexcount** : est un script Python qui permet d'analyser le code source d'un document au format  $\text{Op}\TeX$  pour en extraire en particulier le nombre de mots. Il permet aussi de compter les occurrences des éléments structurants du format  $\text{Op}\TeX$  comme les commandes de sectionnement, les légendes, les tables, les figures, etc. tout cela à la manière de `texcount`.
- nddraw** : fournit des commandes permettant de représenter graphiquement des réseaux de neurones de façon efficaces. Ce package est basé sur `TikZ`.
- book-of-common-prayer** : permet de composer des documents liturgique dans le style du *Book of Common Prayer* de 1979.
- spbmark** : fournit trois commandes `\super`, `\sub` et `\supersub` pour améliorer le positionnement des exposants et des indices, à la fois en mode texte et mathématique. Ces positionnements peuvent aussi être ajustés localement grâce à des paramètres optionnels de ces commandes.

**texdimens** : permet de résoudre certains problèmes concernant l'expression des dimension avec  $\TeX$  dans les différentes unités  $\TeX$ . Il s'agit ici de fournir des commandes comme `\texdiminUNITE{}` où UNITE est une unité (par exemple le package fournit `\texdiminpt{}`).

**ifallfalse** : fournit l'environnement `ifallfalse`<sup>42</sup> et la commande `\orcheck` qui permettent de tester si une chaîne de caractères est contenue dans un autre ensemble de chaîne de caractère, et d'exécuter un code si ce n'est pas le cas. Ce package ne fonctionne pas avec le moteur `lualatex`.

**strands** : fournit des commandes `TikZ` permettant de représenter des opérations sur des ensembles telles que les partitions, les permutations ou bien des *tresses*<sup>43 44</sup>.

code

```

1 \begin{tikzpicture}
2 \strands{p1-n3-e5*v2-t4-f6*s1-n3-p6*n1-e3-s5}
3 \end{tikzpicture}

```

résultat

**inputnormalization** : permet une gestion uniformisée des fichiers d'entrées Unicode pour `LualTeX` et `XlTeX`. En effet, Unicode est un codage particulier car plusieurs caractères peuvent être codés de différentes manières pourtant considérées comme équivalentes. Le problème étant que  $\TeX$  n'est pas au courant de ces codages équivalents, et cela peut poser des problèmes. Unicode définit un mécanisme pour résoudre ces problèmes : les formes normalisées NFC et NFD. Ce package permet d'utiliser les normalisations.

**BasicLaTeX-ru** : est une introduction à  $\LaTeX$  en russe.

**thubeamer** : est un package `beamer` très complet utilisé à l'université Tsinghua, à Pékin.

**ixec** : permet d'exécuter des commandes `shell` depuis votre document et d'en imprimer leurs sorties. Cela suppose évidemment d'utiliser les programmes `pdflatex` ou `latex` avec l'option `--shell-escape`.

**texsurgery** : est un programme python permettant de traiter un source  $\LaTeX$  pour faire évaluer le contenu de quelques commandes et environnements par un noyau `jupiter`, et ainsi disposer des programmes python, sage, R pour réaliser des calculs. Le programme `texsurgery` produit en sortie un nouveau source  $\LaTeX$ , cette fois-ci compilable par les moteurs classiques, et contenant le code produit par l'évaluation par `jupiter` en lieu et place des commandes et environnements dédiés.

**tikz-swigs** : permet de dessiner avec `TikZ` des nœuds ellipsoïdaux divisés en deux.

**newcastle-bst** : est le style `BIBlTeX` Harvard at Newcastle, recommandé par l'université de Newcastle upon Tyne.

42. À l'heure où nous écrivons ces lignes, l'environnement à utiliser est en fait `allfalse`.

43. Notion mathématique ici : [https://fr.wikipedia.org/wiki/Tresse\\_\(mathématiques\)](https://fr.wikipedia.org/wiki/Tresse_(mathématiques)).

44. À l'heure où nous écrivons ces lignes, ce package est bogué mais nous avons fourni à son auteur un patch le corrigeant.

**biblatex-spbasic** : est un style Bib $\LaTeX$  reprenant les caractéristiques de l'ancien style Bib $\TeX$  spbasic.bst. Il est utilisé dans les publications de l'éditeur Springer.

**pyluatex** : permet d'inclure du code python dans un document  $\LaTeX$  et de l'interpréter lors de la compilation! Cela suppose d'utiliser le programme Lua $\TeX$  avec l'option --shell-escape... et c'est un argument supplémentaire pour l'utilisation de ce moteur moderne!

**macrolist** : permet de créer des listes de macros et de les manipuler.

**to-be-determined** : permet de surligner des passages d'un document.

**cartonaugh** : permet de créer des tables de Karnaugh à l'aide de Lua $\TeX$ .

code

```

1 % \begin{texexp}[righthand ratio=0.44]{}
2 \begin{cartonaugh}[4][2][1][\$ab\$][\$c\$]
3 \maxterms{0,2,4,5,6}
4 \minters{3}
5 \terms{1}{\$d\$}
6 \terms{7}{\$d'\$}
7 \implicant{1}{3}
8 \implicant{3}{7}
9 \end{cartonaugh}

```

$ab$	$c$				
		00	01	11	10
0		0	d	1	0
1		0	d'	0	0

résultat

**easing** : est une librairie facilitant l'usage de fonctions mathématiques avec TikZ.

**xistercian** : est un package très complet permettant d'utiliser le système cistercien de notation numérique.

code

```

1 \cisterciannum{2021}

```

résultat

**pdficons** : permet de dessiner des PFD (*process flow diagrams*) voire des PID (*process and instrumentation diagrams*); ce package est dédié à la description graphique des procédés d'ingénierie chimique et possède une documentation très détaillée, dont nous avons tiré la figure page suivante.

**beamertheme-simpledarkblue** : est, comme son nom l'indique, un thème beamer utilisant abondamment la couleur bleue. Il nous vient de Taiwan, pays dont nous ne résistons pas à reproduire ici le drapeau : .

**tonevalue** : permet de représenter graphiquement l'évolution des tonalités d'une langue. De tels graphiques sont utilisés en phonologie, notamment pour les langues parlées en Chine.

**palette** : permet de créer des palettes de couleurs.

**aboensis** : est une fonte reproduisant les caractères manuscrits cursifs utilisés en Scandinavie au xv<sup>e</sup> siècle; son nom nous vient du *Codex Aboensis*, cher à la ville finlandaise de Turku. La très longue documentation de ce package est aussi instructive que magnifique; chose rare dans les documentations de packages, l'une de ses parties,

abondamment illustrée, éclaire le lecteur sur l’histoire de ce codex. Ce remarquable travail de Tommi SYRJÄNEN mérite d’être salué.

code
<pre>1 \abcursivefamily{aboensis}</pre>
résultat

**hvllogos** : permet de créer facilement les différents logos relatifs à T<sub>E</sub>X et à ses logiciels compagnons. Ce package charge les packages fetamont (la police utilisée pour les logos METAPOST et METAFONT), hologo, dantelogo et xspace.

**scrambledenvs** : permettrait l’affichage aléatoire d’indices – lesquels nous ont malheureusement manqué pour mieux vous décrire ce package.

**graphicscache** : diminue le temps de compilation de documents comportant de nombreuses images en utilisant un mécanisme de cache. Par ailleurs, ce package peut attribuer aux différentes images la même résolution. Le tout sans toucher à la commande `\includegraphics`, dont le fonctionnement ne change pas.

**zref-check** : optimise la gestion des références au sein d’un document de manière flexible et configurable.

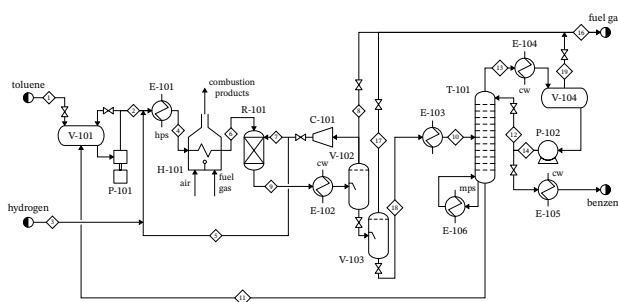
**nwafuthesis** : est une classe permettant de rédiger une thèse au sein de l’université forestière et agricole du Nord-Ouest, sise à Yangling, en Chine.

**smart-eqn** : permet de modifier le style des symboles mathématiques selon un balisage plus léger que les commandes habituelles.

**nchairx** : est un package développé par des professeurs de l’université de Wurtzbourg (Allemagne); il offre un jeu de macros et d’environnements permettant de composer facilement d’élégants documents comprenant des mathématiques.

**tikzbricks** : permet aux grands enfants de représenter des briques de jeux de construction et des volumes réalisés à l’aide de celles-ci. L’idée de ce package est apparue lors de la conférence TUG 2021 !

FIGURE 2 – Un aperçu des possibilités du package pfdicons



## Comment soi-même veiller technologiquement ?

Pour être tenu informé en « temps réel » des nouveautés et mises à jour du CTAN, on peut par exemple consulter régulièrement la page <https://www.ctan.org/ctan-ann> ou, mieux, s’abonner aux flux ou à la liste CTAN-ann qui y sont mentionnés. Il est alors toutefois à noter que, si les nouveautés sont effectivement toutes signalées, les mises à jour ne le sont en revanche que si leurs auteurs ont estimé que l’annonce se justifiait.

## UTILISER DES BOUCLES AVEC TIKZ

*Nous avons découvert cet article de Harald LICHTENSTEIN dans la dernière livraison de Die TeXnische Komödie, la revue de nos homologues germanophones. Sa prose limpide et très pédagogique nous a fait penser qu'il pourrait intéresser un public francophone.*

Lorsque nous dessinons avec TikZ, il est possible d'économiser un temps précieux en utilisant des boucles. Les boucles permettent l'affichage de chiffres ou de lettres. Le présent article explique comment les utiliser.

### Des boucles simples en TikZ avec `\foreach`

Dans le chapitre 2.20 du manuel de TikZ [1] est expliquée la syntaxe des boucles.


code	résultat
<pre> 1  \$\begin{tikzpicture} 2    \foreach \x in {1, 2, 3}{ 3    \node at (\x, 0) {\x} ; 4    } 5  \end{tikzpicture}\$ </pre>	

Ce premier exemple, très simple, nous donne l'image avec les trois nœuds 1, 2, 3.

La syntaxe fonctionne ainsi :

- les lignes 1 & 5 ouvrent et ferment l'environnement TikZ;
- à la ligne 2, la boucle commence par l'instruction `\foreach`; la variable `\x` est déclarée et prendra successivement les valeurs 1, 2 et 3;
- l'accolade de la fin la ligne 2 marque le début des instructions qui seront répétées;
- ces instructions figurent en ligne 3 : pour chacune des valeurs prises par la variable `\x`, l'instruction `\node` va créer un nœud de coordonnées `$(\x, 0)$` dont le contenu, délimité par les accolades, sera `\x`;
- l'accolade de la ligne 4 met un terme à la boucle.

De la même manière, les valeurs prises par la variable peuvent être des lettres :

code	résultat
<pre> 1  \begin{tikzpicture} 2  \foreach \n in {A, B, C} {\$\n\$} ; 3  \end{tikzpicture} </pre>	

Mais dans ce cas, la boucle ne crée pas de nœuds : elle positionne côte-à-côte, sur la même ligne, les différentes valeurs prises par la variable, voilà tout. Les informations fournies à TikZ par ce bref code ne permettent pas de contrôler le positionnement de ces lettres. Par ailleurs, l'affichage des lettres ne fonctionne qu'en mode mathématique : si l'on enlève les balises `$`, rien n'est affiché.

Si la boucle ne doit être répétée que peu de fois, les différentes valeurs prises par la variable peuvent être déclarées explicitement. Mais TikZ peut aussi les déduire des instructions qui lui sont données.

code	résultat
<pre> 1  \begin{tikzpicture} 2  \foreach \n in {A, ..., Z} {\$\n\$} ; 3  \end{tikzpicture} </pre>	

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

code  
résultat

Les trois points (...) déclarent implicitement les valeurs successivement prises par la variable `\n`.

Il arrive que nous ayons besoin de plusieurs variables dans une même boucle. Nous pouvons alors utiliser la syntaxe suivante :

code	résultat
<pre>1  \$\begin{tikzpicture} 2    \foreach \x/\n in {1/A, 2/B, 3/C}{ 3      \node at (\x, 0) {\n} ; 4    } 5  \end{tikzpicture}\$</pre>	A B C

Dans la boucle `\foreach` sont déclarées deux variables, `\x` et `\n`, séparées par le signe `/`. Entre les accolades, les valeurs prises par ces variables sont explicitement déclarées de la même manière : deux valeurs séparées par le signe `/`.

Malheureusement, la déclaration implicite des valeurs ne fonctionne pas pour plusieurs variables déclarées ensemble : il n'est pas possible d'utiliser les trois points pour les déclarer.

```
1  \begin{tikzpicture}
2  \foreach \x/\n in {1/A, .../..., 26/Z}{ << SYNTAXE ERRONÉE
3    \node at (\x, 0) {\n} ;
4  }
5
6  \foreach \x/\n in {1/A, ..., 26/Z}{ << SYNTAXE ERRONÉE
7    \node at (\x, 0) {\n} ;
8  }
9  \end{tikzpicture}
```

## Des boucles complexes en TikZ avec `\foreach`

Une astuce, ou plus exactement une option fournie à l'instruction `\foreach`, nous permet de nous tirer de ce mauvais pas sans avoir à déclarer explicitement chaque paire de valeurs.

```
1  \begin{tikzpicture}[x = 0.35 cm, y = 0.35 cm]
2  \foreach \n [count=\x] in {A, ..., Z}{
3    \node [anchor = base] at (\x, 0) {\n} ;
4  }
5  \end{tikzpicture}
```

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

code

résultat

Quelles nouvelles instructions ont donc été ajoutées ?

- En ligne 1, nous avons réduit l'espacement standard des abscisses : originellement espacées de 1 cm, les valeurs entières sont désormais espacées de 0,35 cm, de manière à ce que les 26 lettres de l'alphabet entrent sur une ligne ;
- en ligne 2, le code devient intéressant : nous introduisons le compteur `[count=\x]`, qui va incrémenter la variable `\x` à chaque boucle. De cette manière, la boucle va

parcourir toutes les lettres de l'alphabet tout en générant les valeurs `\x` nécessaires aux coordonnées des nœuds ;

- en ligne 3, nous passons le paramètre `[anchor = base]` aux nœuds générés par la boucle, de manière à ce que les lettres J et Q, qui en capitales présentent un jambage, ne dépassent pas la hauteur de X ; la position du nœud `(\x, 0)` suit ce paramètre, et la définition du nœud est complétée par son contenu `\n`.

Lors de la rédaction — toujours en cours — d'un livre consacré aux méthodes de cryptographie, nous avons eu recours à une figure carrée formée par les lettres de l'alphabet : la *table de Vigenère*. La composition de celle-ci nous a causé quelques migraines, jusqu'à ce que les boucles TikZ nous fournissent le résultat désiré.

Nous avons utilisé une option offerte par les boucles TikZ : l'entrelacement.

<pre> 1 \begin{tikzpicture}[x = 0.35 cm, y = 0.35 cm] 2 \foreach \m [count=\y] in {Z, ..., A}{ 3   \foreach \n [count=\x] in {\m, ..., Z}{ 4     \node [anchor = base] (P\x\y) at (\x, \y) {\n} ; 5   } 6 } 7 \end{tikzpicture} </pre>	<i>code</i>
<pre> A B C D E F G H I J K L M N O P Q R S T U V W X Y Z B C D E F G H I J K L M N O P Q R S T U V W X Y Z C D E F G H I J K L M N O P Q R S T U V W X Y Z D E F G H I J K L M N O P Q R S T U V W X Y Z E F G H I J K L M N O P Q R S T U V W X Y Z F G H I J K L M N O P Q R S T U V W X Y Z G H I J K L M N O P Q R S T U V W X Y Z H I J K L M N O P Q R S T U V W X Y Z I J K L M N O P Q R S T U V W X Y Z J K L M N O P Q R S T U V W X Y Z K L M N O P Q R S T U V W X Y Z L M N O P Q R S T U V W X Y Z M N O P Q R S T U V W X Y Z N O P Q R S T U V W X Y Z O P Q R S T U V W X Y Z P Q R S T U V W X Y Z Q R S T U V W X Y Z R S T U V W X Y Z S T U V W X Y Z T U V W X Y Z U V W X Y Z V W X Y Z W X Y Z X Y Z Y Z Z </pre>	<i>résultat</i>

Quelles nouvelles instructions ont été maintenant ajoutées ?

- En ligne 2, les valeurs sont incrémentées en sens inverse : TikZ va successivement attribuer à la variable `\m` une lettre de l'alphabet, de Z à A ; cette instruction forme la boucle extérieure ;
- à l'intérieur de celle-ci, en ligne 3, la boucle intérieure définit toujours la variable `\n`, avec la particularité qu'elle hérite la valeur `\m` de la boucle extérieure ; cette valeur est utilisée pour définir l'une des extrémités de la plage que prendront les valeurs successives : en toute logique, la première boucle donnera les valeurs de Z à... Z, la deuxième de Y à Z, etc. ; ces incréments nous montrent que nous approchons du triangle désiré ;



- la troisième ligne définit les nœuds eux-mêmes, avec une particularité : ces nœuds sont nommés au sein des parenthèses ( $P\backslash x\backslash y$ ) précédant le positionnement du nœud<sup>45</sup> ; nous remarquerons que les variables  $\backslash x$  et  $\backslash y$  sont utilisées lors de cette opération de nommage : ainsi, le premier nom sera P11 et le dernier P2626.

Nous obtenons donc une figure comprenant 26 lignes, chacune d'entre elles composée d'une à vingt-six lettres :

- lors de la première boucle,  $\backslash m = Z$ ,  $\backslash y = 1$ ,  $\backslash n = Z$  et  $\backslash x = 1$  ; TikZ va donc composer cette ligne selon l'instruction suivante :

```
1 \node [anchor = base] (P11) at (1, 1) {Z} ;|
```

Il s'agit du Z formant le coin inférieur du triangle de lettres ;

- lors de la dernière boucle,  $\backslash m = A$ ,  $\backslash y = 26$ ,  $\backslash n = Z$  et  $\backslash x = 26$  ; TikZ va donc composer cette ligne selon l'instruction ci-après :

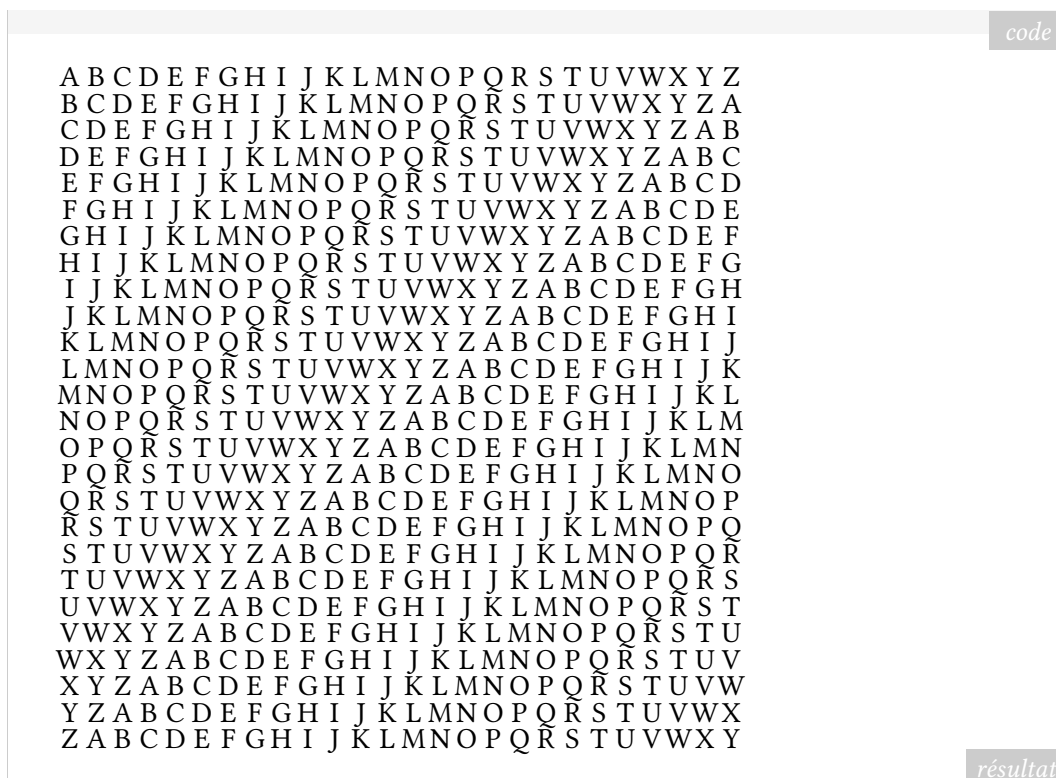
```
1 \node [anchor = base] (P2626) at (26, 26) {Z} ;|
```

Il s'agit du Z formant le coin supérieur droit du triangle.

Pour compléter la table de Vigenère, il nous faut à présent en composer la moitié inférieure gauche.

```
code
1 \begin{tikzpicture}[x = 0.35 cm, y = 0.35 cm]
2 % boucle générant le triangle supérieur
3 \foreach \m [count=\y] in {Z, ..., A}{
4   \foreach \n [count=\x] in {\m, ..., Z}{
5     \node [anchor = base] (P\backslash x\backslash y) at (\backslash x, \backslash y) {\backslash n} ;
6   }
7 }
8
9 % boucle générant le triangle inférieur
10 \foreach \m [count=\y] in {A, ..., Y}{
11   \foreach \n [count=\x] in {\m, ..., A}{
12     \node [anchor = base] (P\backslash x\backslash y) at (27-\backslash x, 26-\backslash y) {\backslash n} ;
13   }
14 }
15 \end{tikzpicture}
```

45. Ce positionnement du nœud est défini par  $at(x, y)$ .



Pour conclure cette étude, quelles nouvelles instructions avons-nous ajoutées ?

De la ligne 9 à la ligne 14, nous avons inséré un nouvel entrelacement de boucles. La boucle extérieure parcourt les lettres de l'alphabet, dans l'ordre alphabétique, mais seulement jusqu'à la lettre Y : en effet, la ligne supérieure de la table, qui se termine par un Z, existe déjà.

La boucle intérieure hérite à nouveau des valeurs de la boucle extérieure et commence par écrire un A sur la deuxième ligne<sup>46</sup> de la figure, en dernière position ; nous obtenons ce résultat par le calcul, les coordonnées étant obtenues grâce à  $(27-\backslash x, 26-\backslash y)$ .

Ainsi, lors de la première itération de la boucle générant le triangle inférieur,  $\backslash m = A$ ,  $\backslash y = 1$ ,  $\backslash n = A$  et  $\backslash x = 1$  ; TikZ va donc composer cette ligne selon l'instruction suivante :

```
1 \node [anchor = base] (P11) at (26, 25) {A} ;
```

Il s'agit du A situé à l'extrémité droite de la deuxième ligne de la table.

Et la dernière itération de la boucle donne les valeurs  $\backslash m = Y$ ,  $\backslash y=25$ ,  $\backslash n = Y$  et  $\backslash x = 1$  ; TikZ va donc composer cette ligne selon l'instruction suivante :

```
1 \node [anchor = base] (P125) at (26, 1) {Y} ;
```

Il s'agit du Y situé à l'extrémité droite de la dernière ligne de la table, qui est désormais complète.

Harald LICHTENSTEIN

46. Rappelons qu'une figure TikZ commence en bas à gauche, au point de coordonnées (0, 0). La ligne inférieure de notre table commence au point de coordonnées (1, 1). Dans notre texte, nous désignons les lignes de la figure par des adjectifs ordinaux, qui sont évidemment utilisés dans le sens de lecture de la langue française : ainsi, la *première* ligne est celle du haut.

La version originale de cet article est parue dans le numéro 2/2021 de *Die T<sub>E</sub>Xnische Komödie*, pages 48 à 52, sous le titre *Schleifen in Tikz*.

Traduction de Patrick BIDEAULT

## Références

- [1] Till TANTAU et al. *The Pgf package. Create PostScript and PDF graphics in TeX*. Version 3.1.9a. URL : <https://ctan.org/pkg/pgf> (visité le 02/07/2021).

## BRÈVES TYPOGRAPHIQUES

### La biennale de Chaumont

Inauguré en 2016 dans la préfecture de la Haute-Marne, le *Signe* de Chaumont est le centre d'art contemporain dédié au design graphique. En plus du Festival international de l'affiche et du graphisme, il accueille désormais une biennale internationale de design graphique, qui a lieu cette année.

Son programme, placé sous le thème de la viralité, est très riche ; c'est évidemment la propagation virale des informations et des images qui est examinée, et plus largement la viralité des représentations. Sont présentées diverses expositions monographiques consacrées à Fanette Mellier, à Michiel Schuurman, au studio iranien Kargah, à Michel Le Petit Didier. Enfin, les écrans du *Design in Motion Festival* d'Amsterdam ont fait le déplacement jusqu'à Chaumont, ce qui permet aux visiteurs de découvrir le meilleur du graphisme en mouvement<sup>47</sup> ; qui sait, peut-être qu'un utilisateur de (All)T<sub>E</sub>X créera une passerelle entre les logiciels que nous utilisons et *DrawBot* ou *p5.js* ! Quoi qu'il en soit, une visite au *Signe* de Chaumont est toujours enrichissante, et cet été plus particulièrement. La biennale dure jusqu'au 21 novembre prochain.

### In memoriam Wolfgang Weingart

Nous apprenons le décès du célèbre typographe allemand Wolfgang Weingart.

Né en février 1941 dans la vallée de Salem, sur les rives allemandes du lac de Constance, où il passe son enfance, il déménage à l'adolescence à Lisbonne, avec ses parents, où il réside entre 1954 et 1956. Ce séjour aura une influence importante sur la vie de ce créateur joyeusement subversif. De retour en Allemagne, c'est à Stuttgart, à la Merz Akademie d'Albrecht Leo Merz qu'il commence en 1958 des études artistiques. L'imprimerie de l'école lui permet découvrir la composition typographique. Il s'essaye également à la création de caractères, sur linoléum et sur bois.

Il travaille ensuite chez Druck + Graphik Wilhelm Ruwe e. K., une imprimerie réputée où il apprend la composition au plomb. Il y fait la connaissance du graphiste de la maison, l'artiste Karl August Hanke (1914 – 1994), qui aura une influence déterminante sur la pratique de Weingart ; il l'encouragea notamment à étudier en Suisse. Après avoir fait à Bâle, en 1963, la connaissance d'Emil Ruder et d'Armin Hoffman, Weingart y déménage en 1964 pour s'établir comme graphiste et y suivre en auditeur libre les cours de la *Schule für Gestaltung Basel* (l'école de design de Bâle, également connue sous le nom de *Kunstgewerbeschule Basel*). Il commence à publier, notamment le très beau *Kinder 1 – Orient Zeichnen – Sonne –*

47. L'auteur de ces lignes a publié un article sur ce festival dans le numéro 85 de *Graphê*, la revue de l'association pour la promotion de l'art typographique

*Moschee*, un livre réalisé avec le pédagogue Volker Merz, qui rassemble des dessins d'enfants du Moyen-Orient. On y remarque déjà un intérêt pour les créations de la jeunesse.

Quelques années plus tard, en 1968, son expérience de typographe et sa créativité ont séduit Armin Hoffman, le directeur de l'école. Il lui propose d'enseigner le graphisme dans un cursus avancé qu'il vient de créer.

Weingart se révèle un enseignant remarquable, non dogmatique, créatif, très à l'écoute de ses étudiants, dont il n'hésite pas à publier les travaux. Son travail de designer graphique, dans lequel il a expérimenté de nombreuses techniques, est exceptionnel. Ce grand professionnel s'est lentement éloigné du style suisse international, dont l'école de Bâle fut le berceau sous l'égide de Ruder. Il a peu à peu trouvé stérile la composition sur grille, qu'il a questionnée, désarçonnant parfois ses étudiants qui venaient précisément à Bâle pour la maîtriser. Mais sa personnalité attachante et rebelle emportait l'adhésion, et il encourageait ses étudiants à exprimer leur personnalité. Il publia leurs travaux dans la revue *Typographic Process*, éditée par l'école et figurant désormais dans les collections du musée d'art moderne de New-York.

Dans son propre travail de designer graphique, il déforma les caractères, les espaça à l'excès, les découpa, allant jusqu'aux limites de la lisibilité. Certaines de ses œuvres sont tout d'abord vues comme des images avant que l'on y remarque les lettres qui les composent, puis le sens de celles-ci. Il comprit l'importance que les ordinateurs allaient prendre dans le secteur du graphisme et de la composition de caractères. Il a fait école bien malgré lui, s'irritant parfois que l'on parle d'un « style Weingart » dès la moindre déconstruction de grille. L'influence indéniable de ce *Swiss Punk*, de ce fondateur de la *New Wave* graphique, dure encore de nos jours ; elle est perceptible chez Anja Kaiser, chez Lamm & Kirch ou chez DeValence, parmi tant d'autres.

Il enseignera jusqu'en 2004, que ce soit dans son école (son enseignement a drainé vers Bâle des étudiants du monde entier) ou, l'été, dans le cursus estival de l'université américaine de Yale, qui se déroulait le plus souvent... en Suisse, à Brissago ! Il a également eu une importante activité de conférencier international. Enfin, il s'est investi dans le comité de rédaction de la revue *Typographische Monatsblätter/Revue suisse de l'imprimerie*, a été membre de l'alliance graphique internationale.

Il laisse un important ouvrage bilingue : *Weingart : Typography*[1]. Et ses étudiants de toutes générations ont compilé ses propos, recueillis lors de son enseignement, en un ouvrage savoureux, *Weingart : The Man and the Machine*[2], dont nous conseillons la lecture à tous les amateurs de design graphique.

Patrick BIDEAULT

## Références

- [1] Wolfgang WEINGART. *Typography. My Way to Typography. Retrospective in Ten Sections / Wege zur Typographie. Ein Rückblick in zehn Teilen*. Baden : Lars Müller Publishers, 2000.
- [2] Susan KNAPP, Michael EPPHEIMER et Dorothea HOFMANN. *Weingart : The Man and the Machine. Statements by 77 of his students at the Basel School of Design (1968–2004)*. Bâle : Karo Publishing, 2014.

## 🌀 LA FONTE DE CE NUMÉRO : LIBERTINUS

Ce numéro de la lettre GUTenberg est composé avec la fonte *Libertinus* qui est un *fork*<sup>48</sup> de la fonte *Linux Libertine*<sup>49</sup>.

### Au début, *Linux Libertine*

#### Présentation générale

*Linux Libertine* est une fonte créée en septembre 2003 par Philipp H. POLL de l'entreprise *Libertine Open Fonts Projekt*. Le but de ce projet est non pas de copier les fontes commerciales *Times Roman* et *Times New Roman* mais d'offrir une alternative professionnelle. Offrir a ici plusieurs sens, puisque la production de cette fonte est faite sous licence GPL<sup>50</sup> et OFL<sup>51</sup> : cela en fait une fonte à sources ouvertes, et gratuite. Elle a d'ailleurs été conçue avec le logiciel libre FontForge<sup>52</sup>, lui aussi sous licence GPL.

Outre le caractère libérateur de cette fonte, celle-ci est très riche puisqu'elle comprend plus de 2000 caractères Unicode et permet de composer les alphabets latin, grec, cyrillique et hébraïque. Elle contient de nombreuses ligatures (voir figure 3), des caractères phonétiques (dictionnaire : dik.sjɔ.nɛʁ), les chiffres elzévirien (0123456789), et des petites capitales.

FIGURE 3 – Quelques exemples de ligatures de la fonte *Linux Libertine* avec les options

Ligatures={Common,Rare,Historic}  
du package `fontspec`.

Th, Qu, fi, ffi, fj, ffl, ct, st

Cette famille de fonte est livrée dans les déclinaisons romane grasse, grasse italique, semi-grasse, semi-grasse italique, régulière et régulière italique. Une variante dite « initiale » est également disponible ; elle est présentée en figure 4.

FIGURE 4 – Illustration de la variante « initiale » de la fonte *Linux Libertine*

W D V B T S

#### Fonte sans serif et monospace

Il existe aussi une fonte sans serif complémentaire à *Linux Libertine* : appelée *Linux Biolinum*, elle reprend les métriques de *Linux Libertine*, dont elle ôte les empattements, comme l'illustre la figure 5.

Elle est aussi l'œuvre de Philipp H. POLL au sein du *Libertine Open Fonts Projekt*. Elle est livrée conjointement avec *Linux Libertine*, dans les déclinaisons régulière, grasse et

48. Un *fork* est un terme anglais signifiant « fourche » ou « bifurcation ». Il désigne un nouveau logiciel créé à partir du code source d'un logiciel existant. Les *forks* sont courants dans le domaine des logiciels libres car leurs licences en permettent la création.

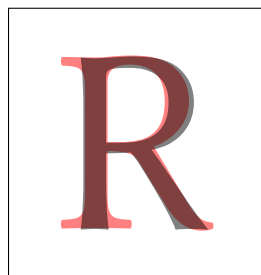
49. Voir <http://libertine-fonts.org/libertine-fonts/>.

50. *GNU General Public License* [anglais] : licence publique générale GNU.

51. *SIL Open Font License* [anglais].

52. <https://fontforge.org/>.

FIGURE 5 – Comparaison du glyphe R avec *Linux Libertine* et *Linux Biolinum*.



italique, mais malheureusement pas grasse italique. La figure 6 illustre la composition d'un paragraphe avec la fonte *Linux Biolinum*.

FIGURE 6 – Composition d'un paragraphe avec la fonte *Linux Biolinum*.

La crise est le moment où l'ancien ordre du monde s'estompe et où le nouveau doit s'imposer en dépit de toutes les résistances et de toutes les contradictions. Cette phase de transition est justement marquée par de nombreuses erreurs et de nombreux tourments.

La fonte *Linux Biolinum* vient aussi avec une très utile variante pour la composition des touches de clavier, comme le montre la figure 7.

Enfin, la famille est complétée par une fonte à chasse fixe qui donne une remarquable cohérence aux documents composés avec la famille *Linux Libertine*.

### Utilisation avec $\LaTeX$

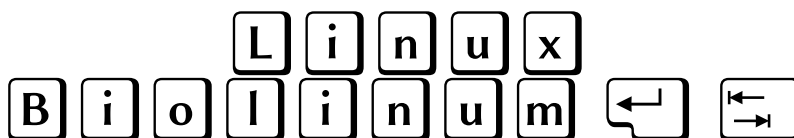
Pour utiliser cette famille de fonte avec  $\LaTeX$ , le package `libertine` a été développé dont les principales fonctionnalités sont utilisables avec  $\LaTeX$ ,  $\text{pdf}\LaTeX$ ,  $\text{Xe}\LaTeX$  ou  $\text{Lua}\LaTeX$ . Avec un simple

```
1 \usepackage{libertine}
```

l'ensemble de cette famille de fontes est chargée. Grâce aux options adéquates, on peut choisir d'utiliser les fontes Type 1 ou bien les fontes OpenType dont tirent profit les moteurs modernes. Il est tout de même à noter que les variantes obliques n'existent pas et qu'elles seront, comme il est d'usage, remplacées par les italiques.

Ce package est utilisable avec les différents moteurs (`tex`, `pdftex`, `xetex` et `luatex`) mais les options ne sont pas toutes utilisables suivant le moteur choisi.

FIGURE 7 – Variante de la fonte *Linux Biolinum* pour la composition de touche de clavier.



## Et de *Linux Libertine* à *Libertinus*

La famille de fontes *Libertinus* a commencé comme un *fork* de *Linux Libertine* et *Linux Biolinum*. Le projet initial a été d'ajouter une fonte OpenType mathématique à cette famille. Avec le temps, ce projet est devenu bien plus qu'un simple ajout du support mathématique, notamment en raison de la correction de certains bogues des fontes *Linux Libertine*. Ce projet a été initié en 2012 par Khaled HOSNY puis, en 2020, il a été repris par Caleb MACLENNAN. Là encore, ce projet est libre sous licence OFL.

*Linux Libertine* et sa compagne sans serif *Linux Biolinum* ont été très utilisées, et une des principales demandes était d'avoir un support mathématique. Il y a eu de nombreuses tentatives, avec plus ou moins de succès, pour trouver des fontes mathématiques qui pouvaient s'intégrer à *Linux Libertine*. En 2012 donc, Khaled HOSNY commença son travail d'ajout du support mathématique à la fonte, en ajoutant une table OpenType MATH ainsi que quelques symboles mathématiques de base. Ce travail a été financé à partir de 2014 par le TUG, et c'est en 2016 que cette fonte a enfin été publiée.

Cette famille de fontes reprend l'ensemble des fontes de *Linux Libertine* : les ensembles roman, sans serif, monospace, ainsi que les variantes initiales et celle pour composer les touches de clavier. Ce à quoi est ajouté la partie mathématique, but premier de ce *fork*.

## Le package **libertinus**

La famille de fonte *Libertinus* peut être très simplement utilisée avec  $\LaTeX$  en chargeant le package d'Herbert VOß :

```
1 \usepackage{libertinus}
```

Ce package est en réalité un emballage des packages du même auteur [libertinus-type1](#) et [libertinus-otf](#). Suivant le moteur utilisé, lors du chargement de [libertinus](#), un de ces packages sera chargé, le *Type 1* si `pdf latex` est utilisé, et le *OpenType* si `xelatex` ou `lualatex` sont utilisés.

Nous nous attarderons ici uniquement sur le package OpenType. L'ensemble des fontes chargées est le suivant :

---

LibertinusSerif-Regular.otf  
 LibertinusSerif-Bold.otf  
 LibertinusSerif-Italic.otf  
 LibertinusSerif-BoldItalic.otf  
 LibertinusSerif-Semibold.otf  
 LibertinusSerif-SemiboldItalic.otf

---

LibertinusSans-Regular.otf  
 LibertinusSans-Bold.otf  
 LibertinusSans-Italic.otf

---

LibertinusKeyboard-Regular.otf  
 LibertinusSerifDisplay-Regular.otf  
 LibertinusSerifInitials-Regular.otf

---

LibertinusMono-Regular.otf

---

LibertinusMath-Regular.otf

---

Pour le détail du paramétrage de [fontspec](#) par le package [libertinus-otf](#), nous vous invitons



à aller lire la documentation. Vous pouvez admirer les différents échantillons de toutes les variantes en figure 8.

FIGURE 8 – Échantillons de la famille de fonte *Libertinus*.

Libertinus Serif Regular AaBbCcDdEeFfGg  
**Libertinus Serif Bold AaBbCcDdEeFfGg**  
*Libertinus Serif Italic AaBbCcDdEeFfGg*  
***Libertinus Serif Bold Italic AaBbCcDdEeFfGg***  
 Libertinus Serif Semibold AaBbCcDdEeFfGg  
***Libertinus Serif Semibold Italic AaBbCcDdEeFfGg***  
 Libertinus Sans Regular AaBbCcDdEeFfGg  
**Libertinus Sans Bold AaBbCcDdEeFfGg**  
*Libertinus Sans Italic AaBbCcDdEeFfGg*  
 L i b e r t i n u s K e y b o a r d ↵ ↲  
 Libertinus Serif Display Regular AaBbCcDdEeFfGg  
**LIBERTINUS SERIF INITIALS REGULAR**  
 Libertinus Mono Regular AaBbCcDdEeFf

Pour illustrer la composition des mathématiques avec cette famille de fonte, nous utilisons l'exemple devenu classique dans le monde de  $\text{\TeX}$  : *le théorème des résidus, ?? page ??ci-dessous*]fig:residus.

FIGURE 9 – Illustration de la composition des mathématiques avec la famille de fonte *Libertinus* à travers l'exemple du théorème des résidus.

**Theorème 1 (Théorème des résidus).** Soit  $f$  une fonction analytique dans la région  $G$  à l'exception des points isolés  $a_1, a_2, \dots, a_m$ . Si  $\gamma$  est une courbe fermée rectifiable dans  $G$  qui ne passe par aucuns des points  $a_k$  et si  $\gamma \approx 0$  dans  $G$  alors

$$\text{Res}_{z=a} f(z) = \frac{1}{2\pi i} \int_C f(z) dz,$$

où  $C \in D \setminus \{a\}$  est une ligne fermée  $n(C, a) = 1$ .

Nous ne ferons pas ici la liste de toutes les options offertes par ces fontes, tant elles sont nombreuses. Le package `libertinus-otf`, pour la version OpenType, offre quelques options de chargement ainsi que quelques commandes permettant de faciliter la gestion des nombreux paramètres de cette famille de fontes, et nous vous invitons à consulter sa documentation. Les paramètres de ces fontes OpenType peuvent être aussi gérés directement avec les outils du package `fontspec`.

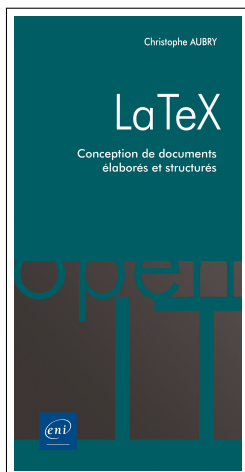
L'ensemble de ces fontes constitue un outil précieux, qui fut en partie financé par la communauté des utilisateurs de  $\text{\TeX}$  et mis à la disposition des amateurs du monde entier, qu'ils utilisent nos logiciels préférés ou d'autres<sup>53</sup>.

Patrick BIDEAULT & Maxime CHUPIN

53. *Horresco referens!*

## COMPTES RENDUS DE LECTURE

### Un vade-mecum



Christophe AUBRY, *LaTeX – Conception de documents élaborés et structurés*, Éditions ENI, juin 2021, ISBN:978-2-409-03081-9, 12 pages, 7 €.

Les éditions ENI, bien connues dans le domaine de l’autoformation en informatique et PAO, ont publié deux ouvrages sur  $\LaTeX$  au printemps 2021, écrits par Christophe AUBRY. Nous commentons ici le plus succinct des deux, en fait une simple feuille de papier glacé, de 21 cm×60 cm, pliée en accordéon à la manière d’une *refcard* (un « pense-bête », en bon français) faite pour tenir dans une poche.

En treize encadrés thématiques (« La structure des documents », « Les tableaux », « Les mathématiques »...), ce dépliant présente un rappel des commandes courantes de  $\LaTeX$ , des options des classes de base et de quelques noms de packages indispensables.

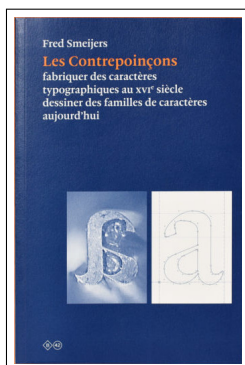
Si vous utilisez  $\LaTeX$  quotidiennement, ces rappels vous seront certainement inutiles. Si vous souhaitez apprendre  $\LaTeX$ , n’espérez pas faire vos premiers pas avec ce seul document. Par contre, si vous êtes utilisateur occasionnel, il pourra vous rafraîchir la mémoire et vous éviter de chercher continuellement dans votre bibliothèque ou sur internet. Un cas d’usage particulier : si vous demandez à un collègue ne connaissant pas  $\LaTeX$  de travailler sur un document que vous avez écrit, cette plaquette lui fera gagner un temps précieux. Propre et moderne, elle rassurera votre collègue davantage qu’un fichier d’aide que vous auriez vous-même préparé.

Le format très compact a manifestement imposé des choix dans ce qui est présenté. Nous aurions aimé y voir une liste des unités de longueur, peut-être aussi le canevas complet d’un document, du `\documentclass` au `\end{document}`, mais globalement, tout y est et le plus gros reproche que l’on puisse faire est sur la forme : ce dépliant n’a manifestement pas été composé avec  $\LaTeX$  !

À part ce détail, la plaquette de Christophe AUBRY mérite vraiment d’être diffusée, car c’est un très bon outil pour que les non-initiés puissent rapidement participer à un travail collectif avec  $\LaTeX$ .

Jérémy JUST

### Un ouvrage historique



Fred SMEIJERS, *Les Contrepointons*, Éditions B42, novembre 2014, traduit par Amarante SZIDON, ISBN:978-2-917-85551-5, 228 pages, 26 €.

Ce livre est ancien (il est paru en 1996 chez Hyphen Press, à Londres) mais il mérite de figurer dans toute bonne bibliothèque consacrée à la typographie.

L’auteur, Fred Smeijers, est un créateur de caractères néerlandais, alors âgé d’une trentaine d’années (il est né en 1961). Typographe chez le fabricant de photocopieurs Océ, il est en butte aux questions de ses collègues ingénieurs, qui ne comprennent pas pourquoi l’impression de caractères nécessite des résolutions aussi précises, et peine à leur fournir des réponses. Il en vient peu à peu à s’intéresser à ses tous premiers prédécesseurs : ceux qui, dès les débuts de l’imprimerie en Europe, gravèrent ces caractères latins que nous utilisons toujours et qui, de par les siècles, ont toujours été une référence commune à tous les créateurs de caractères. Il découvre au musée Plantin-Moretus les poinçons qu’ils utilisaient et se demande comment ils furent fabriqués.

Smeijers a fini par se décider à graver lui-même des caractères, dans une sorte de rétro-ingénierie typographique, pour comprendre leurs formes par la pratique.

Son livre relate ses expériences concrètes de graveur, ce qui en fait un manuel autant qu'un ouvrage historique. Les nombreuses illustrations nous donnent à *voir* les contraintes techniques qui ont présidé à la création des caractères (notamment dans les petits corps, très difficiles à fabriquer) et pesé sur leur apparence finale. L'auteur détaille l'utilisation des limes et burins, compare le champlevage à l'utilisation de contrepointons, revient avec un œil critique sur le *Manuel typographique* de Pierre-Simon Fournier, et détaille la fabrication d'un caractère, du morceau d'acier mis à l'équerre jusqu'à l'impression du caractère sur la page. L'auteur a été jusqu'à mesurer l'épaisseur des copeaux d'acier ôtés par ses burins lors du champlevage, pour montrer aux ingénieurs de l'entreprise la nécessité de résolutions importantes!

L'ouvrage se lit comme un roman et la brève description de la visite au musée Plantin-Moretus du père de l'auteur, professionnel du travail du métal mais peu au fait de la chose typographique, est savoureuse. Mais ce livre, qui procède d'une démarche analytique et concrète, est rigoureux. Ses chapitres introductifs, sur la notion de caractère, sur les trois façons de réaliser des lettres, sur les comparaisons de caractères, sont très pédagogiques et constituent une excellente référence pour les typographes. De par sa pratique de graveur, l'auteur éclaire d'un jour nouveau les origines de l'écriture typographique en caractères latins. Au fil de ces pages, leur lecteur ressent combien la suite des caractères bidimensionnels qu'il parcourt est due à la juxtaposition d'éléments tridimensionnels : ces petites barres d'acier dont l'auteur explique la fabrication et l'utilisation.

Et le remarquable chapitre consacré au jeu du noir et du blanc dans la page, où l'auteur décrit minutieusement la notion de contreforme et son importance dans l'équilibre de la composition, montre bien combien cette question reste actuelle. Car c'est en gravant des poinçons que Smeijers a compris l'importance des contrepointons : ils garantissaient la stabilité des caractères et permettaient à leurs créateurs une productivité raisonnable. Il montre comment le même contrepointon était utilisé pour plusieurs lettres (par exemple le b, le d, le p et le q) et pour plusieurs corps (le contrepointon était taillé obliquement : en l'enfonçant plus ou moins dans le poinçon, on obtenait une forme plus ou moins grande), ce qui explique combien ceux-ci étaient précieux pour les créateurs de caractères. Hélas, seuls les graveurs de poinçons comprenaient la valeur des contrepointons, à la différence des imprimeurs : cela explique que peu de ces contrepointons anciens sont parvenus jusqu'à nous. Le musée Plantin-Moretus n'en compte que 16, contre 4500 poinçons!

L'auteur nous donne des outils permettant de mieux évaluer les caractères typographiques que nous utilisons et questionne avec clarté nos pratiques contemporaines. Comme le dit Smeijers dans sa préface à la nouvelle édition, « les savoirs du passé opèrent comme un véritable miroir permettant de nous évaluer nous-mêmes et de mesurer les progrès techniques accomplis ». Cette phrase suffit à recommander la lecture de cet ouvrage.

Enfin, notons qu'une fonte a été créée par l'auteur pour la nouvelle édition de ce livre, en 2011. Ce caractère, l'Haultin<sup>54</sup>, est d'une rare élégance, à la hauteur de la composition de ce superbe ouvrage. Les lecteurs francophones savent gré aux éditions B42 et à Amarante SZIDON, la traductrice, de le rendre disponible en français.

Patrick BIDEAULT

---

54. Du nom de Pierre Haultin (né en 1510 ou 1513, mort en 1587 ou 1588), graveur de poinçons français. Smeijers a comparé les contreformes des caractères de Haultin avec ses caractères *Quadrat*, récemment créés : il en était fier en raison de leur étroitesse, qui ne nuisait en rien à leur lisibilité. Il a constaté que celles de Haultin, très lisibles, étaient encore plus étroites!

# GUTenberg

Association GUTenberg  
15 rue des Halles – BP 74  
75001 Paris  
France  
secretariat@gutenberg.eu.org

**Site internet :** <https://www.gutenberg.eu.org/>

**Cahiers :** <http://cahiers.gutenberg.eu.org/>

**Problèmes techniques :**

la liste gut : <https://www.gutenberg.eu.org/-Listes-de-diffusion->

le site T<sub>E</sub>Xnique de questions et réponses : <https://texnique.fr/>

la foire aux questions : <https://faq.gutenberg.eu.org>

Cette association est la vôtre : faites-nous part de vos idées, de vos envies, de vos préoccupations à l'adresse [secretariat@gutenberg.eu.org](mailto:secretariat@gutenberg.eu.org).

## Adhésion à l'association

- Les adhésions sont à renouveler en début d'année pour l'année civile.
- Les administrations peuvent joindre un bon de commande revêtu de la signature de la personne responsable; les étudiants doivent joindre un justificatif.

## Tarifs 2021

Les membres de GUTenberg peuvent adhérer à l'association internationale, le TUG<sup>55</sup>, et recevoir son bulletin *TUGboat* à un tarif préférentiel :

**tarif normal :** 65 € (au lieu de 85 \$)

**tarif étudiant :** 40 € (au lieu de 55 \$)

Type d'adhésion	Prix
Membre individuel	30 €
Membre individuel + adhésion TUG	95 €
Membre individuel étudiant/demandeur d'emploi	15 €
Membre individuel étudiant + adhésion TUG	55 €
Association d'étudiants	65 €
Organisme à but non lucratif	130 €
Organisme à but lucratif	229 €

## Règlements

Les règlements peuvent s'effectuer par :

- **virement bancaire**<sup>56</sup> (IBAN : FR76 1870 7000 3003 0191 3568 475)

**Veillez à bien indiquer vos nom et prénom dans les références du virement !**

- Paypal<sup>56</sup> : <https://www.gutenberg.eu.org/?Adherer-en-ligne>
- bulletin et chèque<sup>56</sup> : <https://www.gutenberg.eu.org/?Adherer-a-l-association>

<b>La Lettre GUTenberg</b>	
Bulletin irrégulomestriel de l'association GUTenberg	
<b>Directeur de la publication :</b>	Patrick BIDEAULT
<b>Comité de rédaction :</b>	Patrick BIDEAULT, Denis BRTOUZÉ, Céline CHEVALIER & Maxime CHUPIN
<b>Adresse de la rédaction :</b>	Association GUTenberg 15 rue des Halles – BP 74 75001 Paris
<b>ISSN :</b>	2742-6149 (version numérique)

55. T<sub>E</sub>X User Group [anglais] : groupe (international) d'utilisateurs de T<sub>E</sub>X.

56. Nous vous remercions de privilégier le virement bancaire.