

La L E T R E GUTenberg

Numéro 52 – avril 2024

Compte rendu de la Journée GUTenberg 2023	5
Compte rendu de l'AG 2023	9
Conseil d'administration du 4 février 2024	12
Exposés mensuels	16
Petite histoire de la FAQ L ^A T _E X GUTenberg	20
Fonctionnement de la FAQ L ^A T _E X GUTenberg	33
De nouveaux packages sur le CTAN	46
Lual ^A T _E X et MetaPost au service de la vulgarisation	58
\dante_tutorial:nn{expl3}{2022}	69
Pas à pas : installation, via le réseau, de la T _E X Live 2024	84
Feuille aldine ou... grappe?	90
La fonte de ce numéro : Luciole	100
Comptes rendus de lecture	107
Brèves	109
Rébus	111
Calendrier	111
Acronymes	112
Adhésion à l'association	114

Chers adhérentes, chers adhérents,

réjouissons-nous de faire partie d'une association qui se préoccupe de l'harmonisation des préambules! Cette utile discipline est parfois oubliée, que ce soit dans les relations professionnelles ou internationales, dans la communication, institutionnelle ou non, voire dans la poésie!

Mais à l'association, en novembre dernier, l'harmonisation des préambules

Avez-vous pensé à régler votre cotisation ?

Si vous avez oublié, ce n'est ni trop tard ni difficile :
<https://www.gutenberg-asso.fr/?Adherer-en-ligne>
Voir aussi page 114.

GUTenberg

nous préoccupait grandement¹ : il s'agissait des préambules du code des exemples² de notre nouvelle et merveilleuse FAQ, mise à jour quotidiennement, joyeusement collaborative, au design élégant et si *responsive* que l'on peut la consulter agréablement sur un téléphone portable. Denis Bitouzé nous décrit abondamment ce nouvel outil dans ces pages³.

La mise en service de la FAQ nous a obligé à augmenter fortement la capacité de notre serveur, tout d'abord pour y installer une instance de git et pour qu'il puisse répondre à des demandes en augmentation constante. Cette décision a doublé le coût de location de cette machine. Par ailleurs, nous avons pour l'instant reçu nettement moins de cotisations que l'an dernier à pareille époque : notre situation financière se tend. Si cette année vous n'avez pas encore adressé votre cotisation à l'association, merci de le faire sans tarder.

Vous l'aurez remarqué : cette *Lettre* a une pagination importante. C'est même la *Lettre* la plus longue que l'équipe actuelle ait publiée⁴, avec 114 pages, patiemment rédigées depuis novembre dernier⁵. Mais à vrai dire, il y a eu peu d'activité rédactionnelle à la fin de l'année dernière : notre talentueuse équipe multi-tâches, récemment rejoints par Bastien Dumont et Yannick Tanguy⁶, s'est alors plus concentrée sur la FAQ⁷. Mais depuis lors, divers articles ont étoffé cette livraison de la *Lettre*, dont une contribution reçue... par la poste⁸ : celle⁹ de Jacques André, fondateur de l'association et de la présente publication, qui nous entretient de cette feuille aldine qui marque le titre de nos articles. Mais est-ce vraiment une feuille ? Je vous renvoie à son article en page 90... et je vous encourage à proposer vos contributions pour la *Lettre* en les adressant au secrétariat, qui transmettra à la rédaction !

Heureux que je l'étais de retrouver Jacques à travers sa contribution, je lui ai fait part¹⁰ d'une réflexion : jusqu'à présent, en tête des pages de la *Lettre*, apparaissait le nom de la publication, comme c'est l'usage dans la presse... ou presque : ce nom étant généré par une commande destinée à être utilisée dans le texte courant, il était composé en italiques.

Or l'entête n'est pas du texte courant. Il me semblait donc qu'il devrait être composé en romain. J'avais vérifié dans divers quotidiens, pour trouver essentiellement du romain (voire des logos, mais cet usage est plus

1. Pour cette première note en bas de page d'une *Lettre* qui en comporte beaucoup – 133, à ce que me dit le package `totcount`, de Vasileios Koutavas – veuillez noter que nous aurions pu désigner ce sujet par la *normalisation des préliminaires*, ce qui aurait été nettement moins engageant, si ce n'est franchement calamiteux.

2. Ces exemples sont compilables en ligne, à tout moment !

3. Plus précisément en page 20 et page 33.

4. Comme je l'ai dit lors de l'AG, ce sont rien moins que 500 pages que nous avons publiées dans la *Lettre* à la parution de la *Lettre* 51. Nous avons eu grand plaisir à les publier, et nous espérons qu'elles sont utiles à la communauté francophone.

5. La *Lettre* 51 est parue le 12 novembre 2023.

6. Bastien et Yannick ont été élus lors de l'AG de novembre, dont on trouvera un compte-rendu en page 9.

7. Après avoir harmonisé, nous nous sommes occupés de pages présentant originellement des passages en multicolonne... ou du « W » du lien Wikipedia : votre FAQ, on la bichonne, on la peaufine, on la ripoline.

8. Électronique, la poste en question.

9. Une observation avisée montre que j'aurais mieux fait d'utiliser ici un pluriel. Voir page 108... et encore ailleurs !

10. Par la poste.

récent). Que ce soit en entête ou en pied de page, les quotidiens *L'Aurore*, *Marianne*¹¹, *La Quotidienne*, *Le Temps*, *Paris-Soir*, *Le Figaro*, *Le Monde*, *L'Humanité*, *L'Équipe*, mais aussi *Le Dauphiné* et *Le Parisien*, tous deux originellement libérés, *Ouest-France*, *Les Dernières Nouvelles d'Alsace*, *La Nouvelle République du Centre* et j'en passe, tous composent leur propre titre hors-texte en romain. Et le plus souvent en petites capitales. Devions-nous faire de même ? La réponse de Jacques fut rapide, affirmative mais mesurée : elle était doublée d'un sympathique rappel, selon lequel *les règles typo ne sont jamais formelles !* Nous avons néanmoins composé en romain l'entête en question^{12, 13} et mis en rapport cette dernière réflexion de Jacques avec la syntaxe que nous utilisons : en matière de commandes *LATEX*, comme en toute chose, il importe d'agir avec discernement. Nos outils sont puissants et peuvent aussi générer des défauts.

Vous aurez remarqué le gris typographique de ces pages, qui est dense ; ce n'est pas sans raison ! Il est le fait de Luciole, la fonte que nous utilisons dans ce numéro : nous la décrivons en page 100. C'est une fonte presque unique en son genre, et remarquable à plus d'un titre, mais plutôt que de la commenter en quelques adjectifs, je préfère vous renvoyer à l'article la concernant.

Nous avons commencé à figer le contenu de cette livraison début avril. Depuis lors, il n'a été que peu augmenté¹⁴ : l'essentiel de notre travail fut consacré aux corrections et à la mise en page. Je salue ici le travail de Yann Denichou, membre de l'association, qui nous prête main-forte et corrige patiemment nos erreurs. Son travail est précieux. Sur ce, cette *Lettre* est vraiment trop longue. La rendre publiable nous a pris des jours. Nous espérons ne jamais dépasser à l'avenir son absurde nombre de pages (114 !), qui est vraiment trop important pour notre modeste rédaction. Nous avons même hésité à scinder cette livraison en deux, mais le travail était trop avancé pour recourir à cette solution. Néanmoins, jamais nous n'avons renoncé à nos exigences rédactionnelles : c'est un peu épisés que nous publions cette *Lettre*, mais elle est telle que nous avons envie qu'elle soit. Nous espérons qu'elle vous intéressera et, si quoi que ce soit devait ne pas vous convenir, nous vous invitons à le signaler au secrétariat. Son adresse figure en dernière page.

La vie de l'association suit son cours et de nombreux projets sont en chantier. Ils sont évoqués dans les pages suivantes, mais nous n'avons pas rédigé l'habituel article qui, au risque de la redite, les recense, permettant au lectorat d'avoir une vue d'ensemble des activités de notre association. Nous vous prions de nous excuser pour cette omission ; vous le retrouverez dans la prochaine livraison. Au quotidien, c'est pour les membres du [CA](#) de

11. Oui : jadis, un quotidien portait ce nom.

12. Le lectorat se rappellera que, corrigeant un défaut similaire, nous avions publiquement fait amende honorable en page 27 de la *Lettre* 48 et cessé de composer les patronymes en petites capitales.

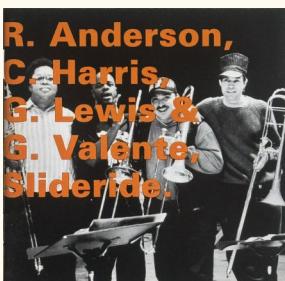
13. Vous constaterez que plusieurs casses sont néanmoins utilisées, notre grand mage de la classe *letgut* ayant gardé en haut de casse l'initiale du premier mot et les trois premières lettres du nom de l'association, ainsi que des chiffres. Est-ce judicieux ? Aurions-nous dû tout mettre en petites capitales ? Tout, y compris les numéros de page ? N'hésitez pas à nous écrire à ce sujet !

14. D'autant qu'il correspondait peu ou prou au contenu d'une *Lettre* idéale tel qu'énoncé dans l'édito de la *Lettre* 46.

nombreux échanges, aussi passionnantes que fructueux¹⁵. Quelques tâches plus astreignantes nous occupent également : François Druel, notre trésorier, s'occupe de demander un rescrit fiscal pour l'association, Maxime Chupin bichonne notre serveur et d'aucuns éliminent des pourriels, ce qui est utile mais peu créatif, et nous prend un temps que nous consacrerions volontiers à d'autres projets. Alors on le dit aux pourriellistes :

- non, l'association GUTenberg, qui ne dispose pas de locaux, n'a pas l'utilité d'un défibrillateur;
- pas plus qu'elle n'a l'utilité d'une fontaine à eau pour ses salariés ; quels salariés ?
- non, elle n'a nul besoin de proposer des [cv](#) de qualité professionnelle aux clients qu'elle n'a pas (et elle recommande de rechercher sur le [CTAN](#) les packages dédiés à cet usage¹⁶);
- pour ce qui est de l'apprentissage de la langue anglaise, ça ira, merci, n'insistez pas ;
- quant au fait que vous proposiez des services d'optimisation pour les moteurs de recherche ou que vous organisiez des safaris, eh bien c'est du pareil au même, et c'est comme pour le dentiste : quand on aura mal au dents, on l'appellera. Inutile de nous contacter avant, merci ;
- ... et on a peine à croire que M. Abramovitch s'intéresse aux destinées de l'association !

Objet	Expéditeur
*****SPAM***** Re:THIS IS ROMAN ABRAMOVICH..CONTACT ME IMMEDIATELY.	waynevas@liv



Une dernière remarque à propos de la composition de ce document : vous le savez, les nombreux trombones présents à la fin de chaque article en contiennent les sources, ce qui est furieusement autotélique¹⁷. Mais voilà : si un article appelle à lui de très nombreux fichiers, c'est une ringuette de trombones qui apparaît. Cela nuit à la mise en pages et nous vous prions de nous en excuser. Une solution [TeXnique](#) sera trouvée ultérieurement¹⁸. D'ici là, et pour patienter, ou alors en compilant¹⁹, n'hésitez pas à écouter le disque ci-contre²⁰, qui montre qu'une ringuette de trombones, si elle peut nuire à la composition d'un document, ne saurait nuire à la musique !

Bonnes compilations !

Patrick Bideault



15. Par exemple, ce sont désormais 614 pages que l'équipe GUT renouveau a publiées dans la *Lettre* depuis qu'elle a redonné vie à l'association en novembre 2020. Plus la FAQ, les exposés mensuels... Nous ne sommes pas peu fiers du travail que nous avons effectué.

16. <https://www.ctan.org/topic/cv>

17. À propos d'autotélique, nous vous renvoyons à la note 86 de l'article consacré à expl3, en page 69.

18. Le lectorat se réjouira que nous ayons remis au lendemain ce que nous aurions pu faire aujourd'hui : à ce compte-là, le présent numéro serait paru en juin !

19. Par exemple avec une [TeX Live 2024](#) installée de fraîche date en suivant les recommandations formulées par Maxime Chupin en page 84 !

20. *Slideride*, par les quatre trombonistes que sont Ray Anderson, Craig Harris, George Lewis et Gary Valente.

⌚ COMPTE RENDU DE LA JOURNÉE GUTENBERG DU 18 NOVEMBRE 2023

La Journée GUTenberg 2023 a eu lieu le 18 novembre à l'École normale supérieure, rue d'Ulm, dans le cinquième arrondissement de Paris. Grâce au directeur du laboratoire de mathématiques de l'établissement, Cyril Imbert, nous avons pu bénéficier de la salle Henri Cartan : un cadre très adapté à notre rendez-vous.

Le programme

Le programme de la journée était le suivant :

09h30-10h00 Accueil et café.

10h00-11h00 Exposé sur [tkz-elements](#) par Alain Matthes.

11h00-12h00 Exposé sur [crossrefenum](#) par Bastien Dumont.

12h00-14h00 Repas de groupe au restaurant Mauzac.

14h00-15h00 Exposé sur la nouvelle foire aux questions ([FAQ](#)) par Denis Bitouzé.

15h00-17h00 Assemblée générale de l'association.

La Journée a été entièrement retransmise en visioconférence sur l'usuelle instance [BBB](#) pour permettre une participation la plus large possible. Nous avons été 25 personnes présentes, et une petite dizaine de personnes en ligne.

Pour les détails de l'assemblée générale, nous renvoyons au compte rendu dédié dans cette *Lettre*.

Les exposés

Cette journée nous a permis de suivre trois exposés d'une très grande qualité. Nous tenons de nouveau à remercier les orateurs pour leur participation.

Alain Matthes : [tkz-elements](#)



C'est Alain Matthes qui, après un petit café d'accueil, a ouvert le bal. Alain est un très grand contributeur dans le monde de [TeX](#), principalement autour de [TikZ](#) et de la géométrie. Beaucoup de ses packages sont reconnaissables à leur préfixe `tkz-` : [tkz-base](#), [tkz-berge](#), [tkz-fct](#), [tkz-graph](#), [tkz-tab](#), etc. Ceux qui nous intéressent ici sont [tkz-euclide](#) et [tkz-elements](#).

Alain est donc venu nous présenter son dernier package. Il s'agit d'une sorte de moteur de géométrie (euclidienne) du plan. S'appuyant sur les possibilités de calcul offertes par LuaTeX, son package permet dans un environnement `tkzelements` de décrire sa figure dans le langage Lua. Ensuite, Alain a conçu une mécanique permettant de transférer les objets calculés dans `tkz-elements` vers son autre package `tkz-euclide` qui, lui, est dédié au tracé de la figure. `tkz-elements` définit les formes géométriques en Lua comme des objets, avec leurs attributs et méthodes : on peut utiliser `point`, `line`, `circle`, `triangle`, `ellipse`, `matrix`, et `vector`. Les calculs internes portent principalement sur les nombres complexes, les barycentres et les objets ; grâce à Lua, ils sont extrêmement précis.

La vidéo est visionnable sur nos chaînes de vidéo à la demande :

- sur Tubedu : <https://tubedu.org/w/hX4c37seEQuNamf8MbXasi>;
- sur YouTube : <https://www.youtube.com/watch?v=vfwMk06ld1Q>.

On ne saurait résister à la tentation de mettre ici un exemple, tant la simplicité du code, la fluidité des descriptions géométriques et la beauté du résultat sont frappants.

Commençons par la déclaration en Lua :

Exemple 1

```

1 \begin{tkzelements}
2   scale = .3
3   z.A = point:new(0, 0)
4   z.B = point:new(6, 0)
5   z.C = point:new(0.8, 4)
6   T.ABC = triangle:new(z.A, z.B, z.C)
7   z.N = T.ABC.eulercenter
8   z.Ea, z.Eb, z.Ec = get_points(T.ABC:feuerbach())
9   z.Ja, z.Jb, z.Jc = get_points(T.ABC:excentral())
10  z.S = T.ABC:spieker_center()
11  C.JaEa = circle:new(z.Ja, z.Ea)
12  C.ortho =
13    circle:radius(z.S, math.sqrt(C.JaEa:power(z.S)))
14  z.a = C.ortho.south
15  C.euler = T.ABC:euler_circle()
16  C.apo = C.ortho:inversion(C.euler)
17  z.O = C.apo.center
18  z.xa, z.xb, z.xc = C.ortho:inversion(z.Ea, z.Eb, z.Ec)
19 \end{tkzelements}
```

Une fois ce code écrit, on utilisera `tkz-euclide` pour organiser notre représentation :

Exemple 2

code

```

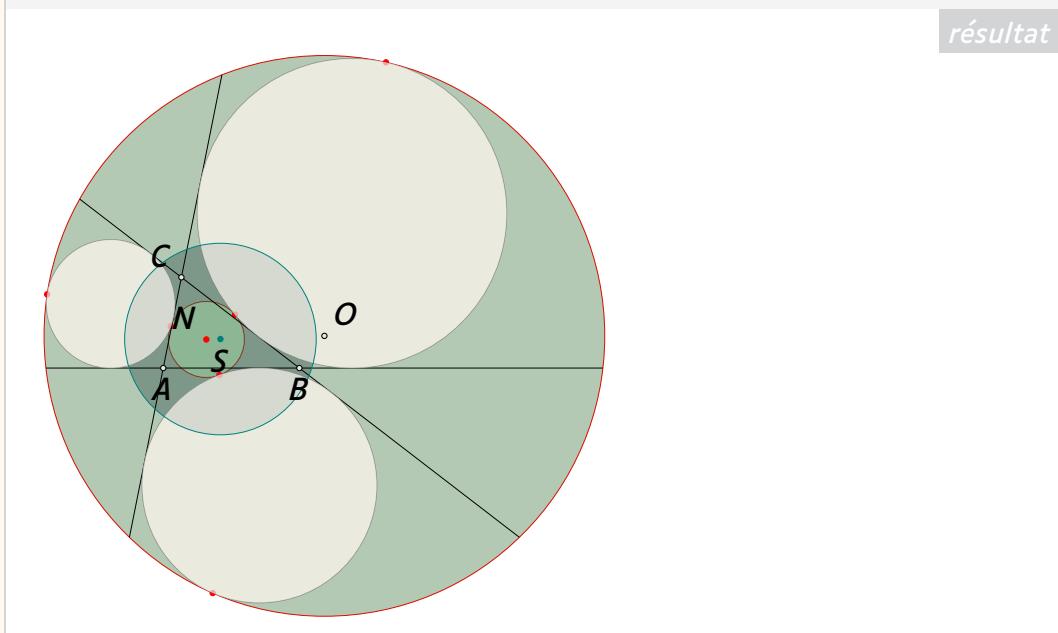
1 \begin{tikzpicture}
2   \tkzGetNodes
3   \tkzDrawCircles[red](0,xa N,Ea)
```

code (suite)

```

4   \tkzFillCircles[green!30!black,opacity=.3](O,xa)
5   \tkzFillCircles[yellow!30,opacity=.7](Ja,Ea,Jb,Eb
Jc,Ec)
6   \tkzFillCircles[teal!30!black,opacity=.3](S,a)
7   \tkzFillCircles[green!30,opacity=.3](N,Ea)
8   \tkzDrawPoints[red](Ea,Eb,Ec,xa,xb,xc,N)
9   \tkzClipCircle(0,xa)
10  \tkzDrawLines[add=3 and 3](A,B A,C B,C)
11  \tkzDrawCircles(Ja,Ea,Jb,Eb,Jc,Ec)
12  \tkzFillCircles[lightgray!30,opacity=.7](Ja,Ea,Jb,
Eb,Jc,Ec)
13  \tkzDrawCircles[teal](S,a)
14  \tkzDrawPoints(A,B,C,O)
15  \tkzDrawPoints[teal](S)
16  \tkzLabelPoints(A,B,S)
17  \tkzLabelPoints[above left](C,N)
18  \tkzLabelPoints[above right](O)
19  \end{tikzpicture}

```

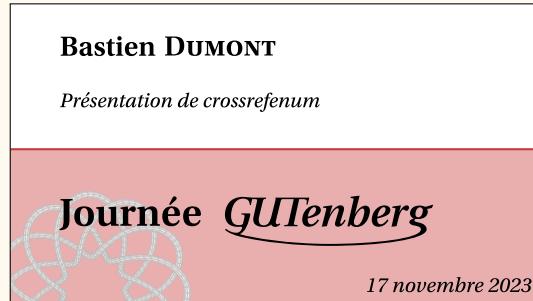


Les diapositives d'Alain Matthes sont disponibles sur la page de notre site web dédiée à la Journée :

<https://www.gutenberg-asso.fr/Journee-GUTenberg-2023>

Rappelons également le site <https://tikz.fr/>, créé par l'association, qui regroupe nombre d'exemples d'Alain.

Bastien Dumont : *crossrefenum*



Bastien Dumont a présenté le fonctionnement de son package *crossrefenum*, qui sert à mettre en forme des énumérations de références internes. D'autres packages *LATeX*, tels que *cleveref*, traitent déjà automatiquement de telles suites de références, convertissant par exemple « p. 1, 2, 3, 5, 5, 9 » en « p. 1-3, 5 et 9 ». Ces packages sont en outre aisément extensibles et personnalisables. Néanmoins, *crossrefenum* répond à plusieurs besoins qu'ils ne permettent pas de satisfaire :

- Supporter les références qui sont formulées selon deux critères à la fois, par exemple « p. 5, n. 4 ; p. 7, n. 2 et 3 »;
- Fonctionner avec n'importe quel format (actuellement, *LATeX* et *ConTeXt* sont supportés) et n'importe quel moteur ;
- Pouvoir être intégré avec n'importe quel système de référencement (par exemple celui de *ConTeXt*, celui du noyau *LATeX*, ou encore *zref*).

Pour atteindre ces objectifs, l'algorithme de *crossrefenum* qui supprime les références redondantes (ci-dessus « 5, 5 ») et crée les intervalles (ci-dessus « 1, 2, 3 » converti en « 1-3 ») est écrit entièrement dans le langage de base de *TeX*, sans reposer sur aucun package extérieur. Pour que *crossrefenum* supporte un type de référence dans un certain format, il suffit de lui indiquer la commande à utiliser pour obtenir la référence brute à partir d'un label et la commande qui introduit la référence dans le document. Par exemple, pour gérer les pages avec *LATeX*, la première sera `\zref@extract{#1}{page}` (requiert *zref*) et la seconde `\pageref{#1}`.

Bien entendu, le package est déjà configuré pour supporter certains types de références :

- *LATeX* et *ConTeXt* : `page`, `note`, `page + note` ;
- *ConTeXt* seulement : `ligne`, `page + ligne` ;
- *LATeX* avec *reledmac* : `edpage`, `edline`, `edpage + edline`.

La vidéo de l'exposé est visionnable sur nos chaînes de vidéo à la demande :

- sur Tubedu : <https://tubedu.org/w/2KD115H6WmT25BtMyH7Bd7> ;
- sur YouTube : <https://www.youtube.com/watch?v=TWYse1JrHNU>.

Denis Bitouzé : Exposé sur la nouvelle version de la FAQ GUTenberg

C'est Denis Bitouzé qui a cloturé la session d'exposés de la Journée. Son exposé a permis de présenter la nouvelle version de la FAQ GUTenberg. Les détails sont à retrouver dans l'article dédié dans cette *Lettre*.

La vidéo de l'exposé est visionnable sur nos chaînes de vidéo à la demande :

- sur Tubedu : <https://tubedu.org/w/nEDK9iLE8UYssphHqtv5y>;
- sur YouTube : <https://www.youtube.com/watch?v=5kB2nfKsx6s>.

Maxime Chupin & Bastien Dumont



⌚ COMPTE RENDU DE L'ASSEMBLÉE GÉNÉRALE DU 18 NOVEMBRE 2023

L'assemblée générale ordinaire de l'association a eu lieu le 18 novembre 2023 à l'[ÉNS](#) lors de notre Journée GUTenberg. L'ordre du jour de celle-ci fut le suivant :

- Bilan moral, discussion et vote
- Point financier avec le rapport financier, la présentation du budget 2024, et la proposition de cotisation pour 2024 (inchangée) avec vote sur chacune des parties
- Sièges à pourvoir au CA : candidatures, discussion et vote
- Discussion sur les cahiers GUTenberg et organisation du financement par souscription
- Discussion sur une possible candidature de GUTenberg pour organisation d'un [TUG](#)
- Questions diverses et libre discussion

Pour permettre la participation la plus large, l'[AG](#) et la Journée entière ont été retransmises en visioconférence. Les votes ont eu lieu de façon hybride par le système de vote électronique Belenios pour les personnes non présentes physiquement sur le lieu de l'AG et en présentiel pour celles présentes physiquement sur le lieu de l'AG (qui pouvaient aussi voter via Belenios).

Patrick Bideault et Maxime Chupin ont présenté le bilan moral que vous pouvez lire dans la *Lettre* 51. Il a été voté à l'unanimité des adhérents présents (21 personnes sur 25 présentes sur place), et à l'unanimité des exprimés sur Belenios (18).

Avec sa faconde habituelle, François Druel a enchaîné avec un point financier comportant un rapport financier, une présentation du budget 2024 ainsi qu'une proposition de cotisation pour 2024. Cela a donné lieu à une discussion sur l'augmentation ou non des cotisations pour suivre l'inflation. Comme notre association, avec ses 88 adhérents pour 2023, a suffisamment de moyens pour réaliser nos actions, les cotisations sont donc reconduites pour 2024. Il a été voté à l'unanimité des adhérents présents (21 personnes sur 25 présentes sur place), et à l'unanimité des exprimés sur Belenios (18).

Nous avons ensuite procédé à l'élection de Bastien Dumont et Yannick Tanguy au [CA](#) de l'association. L'ensemble des participants a salué leur engagement, et nous sommes très heureux de cette élection qui s'est faite l'unanimité des adhérents présents (21 personnes sur 25 présentes sur place), et à l'unanimité moins une voix des exprimés sur Belenios (17).

Discussion sur les *Cahiers* et leurs financement

Jean-Michel Hufflen nous a ensuite fait un point sur la production des *Cahiers* et a ouvert la discussion sur les moyens de financement. Jean-Michel a proposé un financement via souscription et nous a montré que pour l'ancien numéro, pour une production de 100 exemplaires, le coût d'un numéro était de 13 €. Cependant, ces estimations de coût dépendent de nombreux paramètres, notamment conjoncturels, et rien ne garantit le prix pour le prochain numéro. Enfin, la production des numéros ne reposant uniquement sur du temps libre et bénévole, il n'est pas encore envisageable d'assurer une régularité de production, même s'il semble probable d'avoir environ un numéro par an.

Cela a ouvert une discussion très intéressante sur de nombreux aspects, notamment celui de l'objectif des *Cahiers* : est-ce un objet permettant de récolter de l'argent ou bien destiné à faire rayonner le monde \TeX ien francophone ? Évidemment, l'un n'exclut pas l'autre. Plusieurs personnes se sont exprimées pour dire qu'il s'agit avant tout de faire rayonner ($\text{\La}\text{\TeX}$).

Nous avons discuté de l'organisation de la souscription : peut-on faire une souscription en début d'année pour le prochain numéro à venir pour faciliter l'adhésion et la souscription conjointe. La discussion semble avoir montré que cela rend les choses trop compliquées à gérer en général, mais c'est peut-être envisageable pour le prochain numéro : pour pouvoir lancer une impression papier des numéros, il faut avoir une estimation des ventes à venir, et donc que la souscription ait lieu en amont de la parution.

Divers formats de publication ont été envisagés : uniquement numérique ([PDF](#)) gratuit ou payant, impression à la demande, version papier accompagnée d'un [PDF](#) accessible soit dès l'envoi du numéro papier, soit, comme le fait le [TUG](#), à la parution du numéro suivant. Concernant la solution du seul [PDF](#) comme vecteur de publication, une des possibilités est aussi de publier les articles au fil de l'eau, puis de, régulièrement, faire des compilations des articles pour faire un numéro, à imprimer ou non. La discussion a été très riche, même si elle n'a pas permis de déterminer le mode de publication et de financement : en effet, cela repose sur un contexte conjoncturel (sur le prix en particulier), sur la force humaine disponible, sur les préférences

des uns et des autres, etc. Il a été proposé pour éclairer le choix définitif, laissé au CA de l'association, d'organiser un sondage sur diverses solutions possibles.

Discussion sur l'organisation d'une conférence TUG

Maintenant que l'activité de l'association a repris, après des années de mensonge, d'apathie, de langueur et de procrastination, le CA de l'association voulait proposer à la discussion de l'AG la possibilité d'organiser la conférence annuelle du TUG en France.

Le TUG 2023 à Bonn en Allemagne a principalement été organisé par deux personnes. Cette conférence a regroupé environ 80 participants sur 2 jours. Nous ne pourrions l'organiser qu'en 2025 ou 2026, mais il est vraisemblable que ce serait plutôt 2026, les conférences 2023 et 2024 ayant toutes deux eu lieu en Europe. Nous avons discuté, à la lumière des participants à la session 2023, des standards voulu ou non par le TUG. Les participants à Bonn ont notamment trouvé le lieu très (trop?) luxueux. La retransmission en ligne a aussi été assurée par une entreprise extérieure rendant très rigide l'organisation des exposés.

Au moins trois personnes, hors CA, sont motivées par l'organisation d'un tel événement. Nous sommes de plus un certain nombre à travailler dans l'enseignement supérieur et la recherche²¹, milieu dans lequel il est courant d'organiser des conférences. Nous aurons donc assez simplement des lieux potentiels, des contacts et un savoir faire. Il nous reste à faire remonter cette « envie » au TUG et à échanger avec lui sur le cahier des charges !

Tombola grâce à Manuel Pégourié-Gonnard

Pour clôturer l'AG, Manuel Pégourié-Gonnard a fait don à GUTenberg de quelques livres liés à TeX et la typographie. Nous avons donc organisé un tirage au sort parmi les personnes présentes en ligne et en présentiel. Ce fut un moment amusant et convivial. Ces livres vont avoir une nouvelle vie et rendre des personnes heureuses. Nous tenons, de nouveau, à remercier Manuel Pégourié-Gonnard qui fut un membre très actif de l'association, un développeur (L)ATeX impressionnant et un diffuseur très important dans le monde et dans la communauté francophone. Merci Manuel !

Maxime Chupin



21. Ainsi que Didier Verna qui organise régulièrement la conférence Lisp.

⌚ COMPTE RENDU DE LA RÉUNION DU CONSEIL D'ADMINISTRATION DU DIMANCHE 4 FÉVRIER 2024

À vingt heures, en ce dimanche 4 février 2024, la séance du conseil d'administration est ouverte en visioconférence.

Sont présents : Patrick Bideault (président de l'association), Denis Bitouzé (secrétaire), Céline Chevalier (vice-présidente), François Druel (trésorier), Bastien Dumont (administrateur), Jean-Michel Hufflen (administrateur chargé des *Cahiers Gutenberg*), Flora Vern (administratrice), Yannick Tanguy (administrateur).

Sont excusés : Maxime Chupin (secrétaire adjoint), Yvon Henel (trésorier adjoint), Arthur Rosendahl (administrateur).

Le quorum étant atteint, le conseil d'administration est ouvert.

Fonctionnement des listes courriels et groupes de messagerie

Le fonctionnement des listes de discussion/diffusion et des groupes de la messagerie [Signal](#) mis en place et utilisés par l'association est détaillé aux nouveaux arrivés au [CA](#), Bastien Dumont et Yannick Tanguy.

Nous nous assurons que tous les membres du CA sont bien abonnés aux listes les concernant.

Les décisions concernant les exposés Gutenberg étaient jusqu'alors prises au sein du bureau alors que d'autres administrateurs peuvent être intéressés. La création d'un canal dédié à ce sujet est discutée et, plutôt qu'un groupe Signal, le choix se porte sur une liste de diffusion par courriel.

Cahiers

Jean-Michel présente le prochain numéro (n° 59). Il sera peut-être partiellement en couleur, notamment pour l'article de Thomas Savary. Il demandera un devis à l'imprimeur dès que le numéro sera finalisé et une souscription sera alors lancée (nous renonçons pour l'instant aux abonnements), ce qui va faire l'objet d'un article dans la prochaine *Lettre*.

Concernant les frais postaux, il convient de ne pas dépasser 130 pages.

Sont ensuite discutées la mise en ligne du n° 58 sur le site des *Cahiers* et la récupération de la gestion des [DOI](#) des anciens numéros.

Organisation du congrès annuel 2026 du [TUG](#)

L'association souhaiterait organiser le congrès annuel du [TUG](#) en 2026. Nous allons l'annoncer officiellement au [TUG](#) et lui demander des informations.

François met en garde sur la potentielle importante mobilisation de fonds que cela peut occasionner.

Lors de la Journée GUTenberg 2024, Didier Verna avait fait part de son expérience d'organisateur du symposium Lisp et nous avait proposé de nous faire bénéficier de son expertise.

Il nous faut trouver et choisir un lieu : Paris ? Ou la quiétude provinciale : Dunkerque, Lille, Strasbourg, le viaduc des Fades etc. ?

Nous allons chercher un partenariat avec une université de façon à organiser les conférences dans ses locaux et ainsi minimiser les frais pour le TUG, pour GUTenberg et aussi, surtout, pour les participants. Cela nécessite d'avoir dans cette université un relais prêt à s'investir.

Bilan d'étape de la FAQ

Denis remercie :

- Yannick pour ses nombreuses contributions au fond ;
- Bastien pour l'infrastructure²² ;

de la FAQ.

Denis a rapporté des bogues aux mainteneurs du thème [Sphinx PyData Sphinx Theme](#) utilisé pour la FAQ et qui souffre actuellement de quelques défauts.

Denis a pris contact avec Jean-François Burnol qui est le principal mainteneur de la sortie PDF via L^AT_EX de documentation conçues avec Sphinx car, si une telle sortie fonctionne bien avec la FAQ, elle ne peut pour l'instant inclure les résultats des exemples L^AT_EX « éditables et compilables » qu'on peut voir sur le site Web. Jean-François a une piste pour améliorer cette situation, mais n'a pas le temps d'y travailler actuellement.

Le projet de sources de la FAQ pas seulement en Markdown mais aussi en L^AT_EX n'est pas abandonné. Il n'a cependant n'a pas encore débuté.

Denis va proposer au TUGboat un article sur la migration de la FAQ.

Bilan d'étape de GitLab

Nous rencontrons des soucis de place sur le disque dur du serveur qui perd quotidiennement plusieurs GB. Il semble que cela soit dû à des instances Docker de GitLab que nous devons manuellement supprimer. Nous ne comprenons pas d'où cela vient et allons demander de l'aide sur la liste GUTenberg dédiée au serveur de l'association.

Par ailleurs, nous souhaitons à terme proposer l'hébergement de projets T_EX francophones ainsi que le référencement des ressources francophones (avec base de données).

Au sujet du serveur, nous cherchons un expert Spamassassin car certaines de nos listes sont assez massivement spammées et supprimer les spams passés entre les mailles du filet reste franchement peu pratique sur notre système d'archives [Hyperkitty](#) !

22. Notamment pour les scripts qu'il a conçus et qui améliorent significativement le rendu de la FAQ.

Bilan d'étape des exposés

Les exposés que nous organisons sont salués pour leur qualité, et l'association a souvent été félicitée pour cette initiative. Merci aux différents intervenants qui ont tous proposé des contenus de grande qualité!

Nous espérons qu'ils vont susciter des adhésions supplémentaires²³. Pour favoriser cela, nous envisageons d'appeler à cotisation sur le transparent d'accueil de notre instance de visioconférence, ainsi qu'en signature des courriels d'annonce et de rappel des exposés.

Les prochains orateurs prévus sont, en mars, Thomas Savary (suite de son premier exposé sur la typographie et le package [lua-typo](#)) et, en avril, Paul Gaborit (sur le package [tabulararray](#)).

Nous réaffirmons que les thématiques des exposés sont larges : (L)A_TE_X, logiciels annexes, typographie, etc.

Cotisations

Désormais, les adhérents peuvent cotiser via [Stripe](#) qui est plus pratique et moins cher que [Paypal](#).

François a entamé la démarche administrative demandant la reconnaissance d'intérêt général.

Journée 2024

François propose de contacter Ève Gilles, Miss France et étudiante en mathématiques, pour lui demander de parrainer la Journée 2024.

Comme en 2023, nous proposons que cette Journée ait lieu à l'automne, de préférence en novembre. Parmi les thèmes envisagés : le guillochis, L_AT_EX3, accessibilité des PDF.

Archives GUTenberg et dépôt [CTAN](#)

Rien n'a été fait depuis notre « décision » de continuer les pourparlers pour que Jérémy Just restitue à l'association les dépôts qui lui sont relatifs. Nous constatons que n'avons pas avancé sur ce point-là.

Discourse

Comme nous l'avons déjà annoncé, nous souhaitons faire migrer la vénérable liste de discussion gut@ens.fr vers un serveur [Discourse](#). Nous rappelons qu'il permettra à ceux qui le souhaitent de l'utiliser comme une simple liste de discussion.

Une telle migration, notamment des archives de la liste, n'est pas une mince affaire. Heureusement, [une telle entreprise a été effectuée](#) pour la liste

23. ... qu'il s'agissent d'adhésions nouvelles ou de réadhésions. Francophones de tous les pays, unissez-vous pour utiliser (All)T_EX!

francophone de [Lilypond](#) et la personne qui s'en est chargée, Jean Abou Samra, a gentiment accepté de nous faire profiter de son expérience.

Lettre

Patrick a commencé à traduire l'article \dante_tutorial:nn{expl3}{2022} sur [LATEX3](#) parue dans l'édition du 4^e semestre de *Die Texnische Komödie*²⁴.

L'article sur la migration de la FAQ que Denis proposera au *tugboat* paraîtra préalablement dans la *Lettre*.

Denis et Maxime proposent de traduire l'article du blog « [LATEX3 : Programming in LATEX with Ease](#) ».

François propose un concours de code de « la plus belle réalisation » en (I)TEX avec une contrainte légère, par exemple 25 ou 50 lignes (d'au plus 80 caractères) de code. Afin que tous les participants puissent partir sur les mêmes bases, ce concours serait lancé après sortie de la *TeX Live 2024*.

Points divers

- L'association va adhérer à l'[APRIL](#)²⁵, ce qui représente une dépense de 50 euros.
- Il y a un certain nombre de salons auxquels nous pourrions participer :
 - [Journées réseau de l'enseignement et de la recherche](#) du 10 au 13 décembre 2024 à Rennes.
 - [Journées du libre éducatif](#) le 29 mars 2024 à la faculté de droit de l'Université Paris-Est Créteil : exposé avec Cédric Pierquet et Christophe Poulain, auquel Maxime se joint.
 - [Open Source Experience](#), anciennement *Paris Open Source Summit* : faire acte candidature pour éventuellement bénéficier d'un stand pas hors de prix, voire gratuit ?

Pour tous ces salons, nous sollicitons les adhérents et sympathisants afin de nous aider (organisation, permanences, etc.).

- Il est possible que <https://linuxfr.org/> relaie nos annonces. François se charge de prendre contact.
- À ne pas oublier :
 - le dépôt légal *Cahiers* ;
 - le dépôt des [PV](#) d'[AG](#) en préfecture ;
- Yannick poursuit son projet TransLATEX de traduction en français d'extensions courantes de LATEX. Il le basculera bientôt sur l'instance GitLab de l'association et écrira à son sujet un article dans la *Lettre*.

La réunion prend fin vers 22:30.

Patrick Bideault & Denis Bitouzé



24. Cet article est à retrouver en page [69](#).

25. L'APRIL est l'association pour la promotion et la recherche en informatique libre. Voir <https://www.april.org/>.

EXPOSÉS MENSUELS SUR (LA)TEX ET AUTRES LOGICIELS

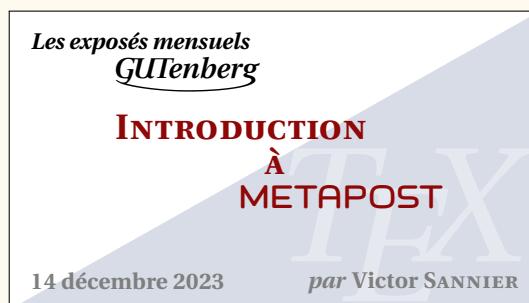
Désormais, les exposés en visio-conférence GUTenberg sont un rendez-vous mensuel ancrée dans la vie de l'association. Il est donc naturel d'en rendre compte dans cette *Lettre*.

Exposés récents

Depuis la *Lettre* 51, il y a eu cinq exposés.

Introduction à MetaPost

Le jeudi 14 décembre 2023, à 20 heures, Victor Sannier nous a proposé un exposé d'introduction à MetaPost.



Nous rappelons qu'au sein de son projet ruscap, Victor Sannier recrée avec METAFONT une écriture manuscrite ancienne, la capitale rustique²⁶. Une prochaine *Lettre* sera consacrée aux fontes anciennes et à leur re-création par l'informatique.

Vous pourrez trouver plus de détails sur la page du site de l'association dédiée à cet exposé :

<https://www.gutenberg-asso.fr/14-decembre-Expose-sur-MetaPost>

Vous pourrez notamment y trouver les diapositives utilisées. Par ailleurs, nous avons enregistré l'exposé et l'avons mis en ligne sur nos deux canaux de vidéo à la demande :

<https://youtu.be/JiQLOZTdFnI>
<https://tubedu.org/w/sSqPHMrVLrL78orCNQVehw>

L'algorithme de Knuth-Plass

Le jeudi 11 janvier 2024, à 20h, nous avons eu le plaisir d'écouter Didier Verna nous parler de l'algorithme de Knuth-Plass. Un exposé d'une certaine longueur et ô combien passionnant!

26. [https://fr.wikipedia.org/wiki/Rustica_\(écriture\)](https://fr.wikipedia.org/wiki/Rustica_(écriture))



Des informations supplémentaires sont proposées sur la page du site de l'association dédiée à cet exposé :

<https://www.gutenberg-asso.fr/11-janvier-2024-Expose-sur-l-algorithme-de-Knuth-Plass>

Vous pourrez notamment y trouver son support de présentation.

Nous avons enregistré l'exposé et l'avons mis en ligne sur nos deux canaux de vidéo à la demande :

<https://www.youtube.com/watch?v=48JNXcaKj2A>
<https://tubedu.org/w/ghJqXvVKy9hk5gkniSNUM8>

Exposé sur LATEX et R

Le jeudi 1^{er} février 2024, nous avons écouté Philippe Michel nous parler de son utilisation du logiciel R conjointe à l'utilisation de LATEX. Nous saluons la limpidité du propos de Philippe et l'efficacité des solutions proposées, dont le rendu est remarquable.



Plus d'informations sont disponibles sur la page du site de l'association dédiée à cet exposé :

<https://www.gutenberg-asso.fr/1er-fevrier-2024-Expose-sur-LaTeX-et-R>

Nous avons, là encore, enregistré l'exposé et l'avons mis en ligne sur nos deux canaux de vidéo à la demande :

<https://www.youtube.com/watch?v=t08rbY64Jj0>
<https://tubedu.org/w/9j5XJHP1grkKWzydfzyZwM>

Composer comme un pro avec lua-typo

Le jeudi 7 mars 2024, pour la deuxième fois de la saison, nous avons écouté Thomas Savary. Dans le prolongement de son premier et passionnant exposé

sur [lua-typo](#)²⁷, il nous a montré comment corriger en pratique les problèmes révélés par le très utile package [lua-typo](#) de Daniel Flipo.

Plus d'informations sont disponibles sur la page du site de l'association dédiée à cet exposé :

[https://www.gutenberg-asso.fr/7-mars-2024-Expose-intitule
e-Composer-comme-un-pro](https://www.gutenberg-asso.fr/7-mars-2024-Expose-intitule-Composer-comme-un-pro)

Le montage vidéo de cette intéressante conférence n'a pas encore été fait, et sera mis en ligne dès que possible sur cette même page.

Exposé sur le package `tabulararray`

Le jeudi 4 avril 2024, nous avons écouté Paul Gaborit nous présenter le très efficace package [tabulararray](#).

Rappelons que [tabulararray](#), de Jianrui Lyu, facilite grandement la composition et améliore le rendu des tableaux, et s'impose peu à peu comme un standard. Une preuve supplémentaire de l'utilité de la couche de programmation de [LATEX3](#) !

Plus d'informations sont disponibles sur la page du site de l'association dédiée à cet exposé :

[https://www.gutenberg-asso.fr/4-avril-2024-Expose-sur-le-p
ackage-tabulararray](https://www.gutenberg-asso.fr/4-avril-2024-Expose-sur-le-package-tabulararray)

Le montage vidéo n'a pas encore été fait, et sera mis en ligne dès que possible sur cette même page.

Exposés à venir

Le jeudi 2 mai, exposé de Bastien Dumont et Yannick Tanguy consacré au fonctionnement de la nouvelle version de la [FAQ LATEX](#) GUTenberg du point de vue du contributeur.

Le 6 juin, Julien Labbé nous présentera son package [overarrows](#).

Projet participatif

Pour rappel, l'association GUTenberg organise des exposés mensuels sur des sujets bien entendu connexes à [\(L\)ATEX](#). Il pourra y être question :

- de la gestion de tel ou tel matériel (tableaux, formules de mathématiques, flottants, bibliographie, etc.);
- d'utilisation de packages ou de classes;
- de création de packages ou de classes;
- de programmation, notamment en [LATEX3](#);
- de typographie;
- de création de dessins;
- de polices de caractères;

27. Le premier exposé de Thomas est disponible sur le site de l'association, à l'adresse <https://www.gutenberg-asso.fr/14-septembre-Expose-sur-lua-typo>.

- d'éditeurs de texte ;
- de formats autres que L^AT_EX ;
- de logiciels de gestion de version (par exemple [Git](#)) ;
- etc.

... et ce, sur des thématiques souvent généralistes, appartenant aux sciences humaines comme aux sciences dures. Ces exposés, en visio-conférence, sont a priori une durée d'une heure au plus, questions comprises.

Lorsque nous avons annoncé l'organisation de ces exposés, nous avons bien entendu lancé un appel aux orateurs volontaires (si possible n'étant pas administrateurs de l'association afin de laisser la place à d'autres). Toutes suggestions (exposés de vous-mêmes ou d'autres personnes, sujets, etc.) sont les bienvenues et sont à adresser au secrétariat de l'association (secretariat@gutenberg-asso.fr).

L'ensemble des informations (programme passé, futur, et informations de connexion) sont accessibles sur la partie dédiée de notre site internet :

<https://www.gutenberg-asso.fr/-Exposes-mensuels->

Agenda numérique

Pour rester facilement informé et ne manquer ainsi aucun exposé, nous avons mis en place un *framagenda* pour les évènements GUTenberg accessible à l'adresse :

<https://framagenda.org/apps/calendar/p/DnkaimkPTyBDsQ7z/dayGridMonth/2023-09-01>

qui fournit (en cliquant sur les paramètres de l'agenda) un lien webcal pour la synchronisation avec vos agendas personnels :

<webcal://framagenda.org/remote.php/dav/public-calendars/DnkaimkPTyBDsQ7z/?export>

Maxime Chupin & Patrick Bideault



PETITE HISTOIRE DE LA FAQ LATEX GUTENBERG

Résumé

L'objet de cet article est de retracer l'historique de la foire aux questions ([FAQ](#))²⁸ LATEX GUTenberg.

Origine

Les premières traces de questions (L)ATEX fréquemment posées, avec leurs réponses, se trouvent sur le forum *Usenet anglophone comp.text.tex*²⁹ sous la forme de *posts* intitulés *TeX, LATEX, etc. : Frequently Asked Questions with Answers [monthly]*, que Bobby Bodenheimer a commencé à publier en 1990 et qu'il a progressivement enrichis.

Les premières traces de FAQ (L)ATEX francophone remontent à 1992 sous la forme d'un article des *Cahiers* [1] qui était une traduction française par Philippe Louarn (aidé de Jacques André, Bernard Gaule et Éric Picheral) du *post* de Bobby Bodenheimer en date du 2 mars 1992. Une deuxième édition de cet article, [2], traduite par les membres de l'époque du bureau de l'association GUTenberg, est parue en avril 1996 lorsque la maintenance de la FAQ de Bobby Bodenheimer a été prise en charge par le [TUG](#) britannique UK-TUG.

En 1996, Fabien Torre a produit et hébergé, sous la forme d'un site Web (quasi) statique, une version [HTML](#) de cette dernière FAQ, connue sous le nom de [FAQ « GRAPPA »](#)³⁰. Celle-ci est encore disponible à l'adresse <https://faqlatex.fabien-torre.fr/>.

Parallèlement à cela, Marie-Paule Kluth a publié la FAQ en 1999 chez Vuibert [3]. Cet ouvrage bénéficia d'une réédition l'année suivante [4].

Ensuite, et jusqu'en 2004, un projet de refonte complète de la FAQ a été mené par Benjamin Bayart avec l'aide de nombreux contributeurs (dont Yvon Henel, trésorier adjoint actuel de GUTenberg). L'ambition était notamment de fournir une version [PDF](#) de cette FAQ. Les défis techniques que cela impliquait, outre l'absence d'outils efficaces de travail collaboratif, semblent avoir eu raison de la survie du projet.

FAQ GUTenberg (2013–2020) : Dokuwiki

À partir de 2013, Jérémy Just (alors président de GUTenberg), avec l'aide de Maxime Chupin (secrétaire adjoint actuel de GUTenberg) et Manuel Pégourié-Gonnard, migre la FAQ de Benjamin Bayart sur système [Dokuwiki](#) : l'idée d'une version [PDF](#) est abandonnée au profit de celle d'un site Web dynamique, simplifiant notamment le travail collaboratif des contributeurs. Jérémy Just y ajoute les questions/réponses de la FAQ (L)ATEX anglophone et

28. Traduction libre habituelle de *Frequently Asked Questions* qui, contrairement à la traduction formelle « questions fréquemment posées », conserve l'acronyme « FAQ ».

29. Les archives jusqu'au 22 février 2024 en sont conservées à l'adresse <https://groups.google.com/g/comp.text.tex>.

30. « GRAPPA » pour le *Groupe de recherche en Apprentissage Automatique*, au sein de l'université de Lille.

commence progressivement à les traduire. Il est rejoint dans l'aventure par 7 personnes dont (chronologiquement) :

- Yvon Henel;
- moi-même, Denis Bitouzé, auteur du présent article³¹ ;
- Yannick Tanguy;
- Bastien Dumont.

Le travail accompli par Jérémie Just, remarquable, est à saluer. Tout d'abord techniquement, l'adaptation de [Dokuwiki](#) aux besoins d'une FAQ [LaTeX](#) n'a certainement pas été une mince affaire. Ensuite, particulièrement grâce à la contribution essentielle de Yannick Tanguy, un nombre incalculable de questions/réponses ont été améliorées, actualisées, traduites mais aussi créées³² et ce, de façon à la fois significative et pertinente.

Dès la parution de cette FAQ [Dokuwiki](#), j'ai voulu y contribuer mais ai été rebuté par plusieurs aspects de ce site et du système sous-jacent :

- son design général tel qu'on peut le voir [figure 1](#);

FIGURE 1 – Exemple de page de la FAQ sous Dokuwiki (vue ordinaire)

The screenshot shows a DokuWiki FAQ page with the following details:

- Title:** [[Comment faire un tableau?]]
- Page Header:** Vous êtes ici : Bienvenue sur la FAQ LaTeX francophone » 3. Composition de documents » Tableaux » Comment faire un tableau?
- Page Navigation:** Afficher le texte source, Rechercher, Administre, Derniers changements, Plan du site, S'identifier.
- Content Area:**
 - Section:** Comment faire un tableau?
 - Description:** L'environnement permettant de faire un tableau en mode texte s'appelle tabular. Il prend un argument optionnel et un argument obligatoire.
 - Code Example:**

```
\begin{tabular}{(position)}{(format)}
    contenu du tableau
\end{tabular}
```
 - Text:** L'argument obligatoire (format) décrit le format des colonnes du tableau. Cela suppose que vous connaissiez à l'avance le nombre de colonnes que vous allez produire, sinon TeX affichera un message d'erreur. Une colonne est représentée par une lettre parmi:
 - l (pour que le texte soit aligné à gauche dans la colonne),
 - c (pour que le texte soit centré dans la colonne) et
 - r (pour que le texte soit aligné à droite).
 - Text:** Les trois types de colonne (l, c et r) ne passent jamais à la ligne et sont, par conséquent, adaptées aux cellules dont le contenu est court. Lorsque les cellules sont bien remplies, il faut opter pour p{larg} (p comme *paragraphe*) qui compose le contenu de la cellule dans un paragraphe justifié de largeur *larg*.
 - Text:** Une répétition *n* fois d'un motif élémentaire est facilitée par *{n}{motif}. *{20}{lcc} est équivalent à lccclccclcc...lcc (en tout 60 colonnes), plus concrètement l'exemple suivant montre un tableau utilisant ce genre de descriptions.
 - Text:** Exemple de tableau (remarquez que le second | fait partie de l'argument de *).

0	1	2	3	4
1	2	3	4	5
2	3	4	5	6
 - Text:** L'argument optionnel (position) définit la position verticale du tableau par rapport à la ligne courante: t pour top (le haut du

- son caractère non ou, désormais, piétinement réactif³³, tel qu'on peut le voir [figure 2](#) page suivante;
- l'ergonomie du site. Par exemple la barre de recherche n'est accessible que si l'on se trouve en haut d'une page (cf. capture d'écran de droite de la [figure 2](#) page suivante);

31. Comme expliqué ci-après, significativement seulement depuis 2020 (mais assez peu sur le contenu).

32. Il y a actuellement plus de 1200 questions/réponses !

33. Un site Web réactif (en anglais, *responsive web design*) permet aux pages Web de modifier leur disposition et leur apparence pour s'adapter à différentes largeurs d'écran, résolutions, etc. Cf. https://developer.mozilla.org/fr/docs/Learn/CSS/CSS_layout/Responsive_Design.

FIGURE 2 – Exemple de page de la FAQ sous Dokuwiki (vue téléphone portable)

The screenshot shows a mobile phone displaying a Dokuwiki FAQ page. The header reads "[[Comment faire un tableau?]] FAQ LATEX". The page content is as follows:

Vous êtes ici : Bienvenue sur la FAQ LaTeX francophone » 3. Composition de documents » Tableaux » Comment faire un tableau?

Navigation menu:

- Afficher le texte source
- Administre
- Derniers changements
- Plan du site S'identifier

Search bar: Rechercher

Table of contents (partially visible):

- 1. Général
- 2. Programmes
- 3. Composition de documents
 - A
 - D
 - F
 - II
 - L
 - T

Text content:

Comment faire un tableau?

➤ L'environnement permettant de faire un tableau en mode texte s'appelle tabular. Il prend un argument optionnel et un argument obligatoire:

```
\begin{tabular}[<position>]{<format>}
  contenu du tableau
\end{tabular}
```

L'argument obligatoire *<format>* décrit le format des

- l'éditeur en ligne **Dokuwiki** permettant de contribuer à la **FAQ** est très peu efficace et ergonomique, notamment par rapport aux éditeurs puissants (tels qu'Emacs) dont on peut disposer en local sur sa machine;
- la **syntaxe Dokuwiki**, très lourde;
- l'impossibilité de faire des modifications en masse (éventuellement scriptables), par exemple pour remplacer 180 occurrences dans 74 fichiers de :

Exemple 3

```

1 \documentclass{...}
2 ...
3 \usepackage[french]{babel}
```

par :

Exemple 4

```

1 \documentclass[french]{...}
2 ...
3 \usepackage{babel}
```

Comme Jérémy Just refusait d'envisager toute autre solution technique, je n'ai quasiment pas contribué à cette FAQ et m'en suis désintéressé. En novembre 2020, après mon élection au CA de GUTenberg, j'ai fait part de mon constat qui a été partagé par la plupart des administrateurs et, comme j'ai eu l'espoir d'être mieux entendu par Jérémy Just, j'ai commencé à prospecter d'autres solutions, avec comme prérequis que :

- les sources de la FAQ soient en syntaxe populaire, si possible... \LaTeX !
- les sources soient stockables sur un dépôt [Git](#)³⁴ afin de, sinon faciliter, du moins rendre très efficace le travail collaboratif;
- ce dépôt [Git](#) soit public afin que chaque (potentiel) contributeur puisse le cloner pour pouvoir le modifier dans son environnement de travail habituel et, au besoin, en masse;
- un moteur de recherche soit *intégré*, notamment pour ne pas devoir recourir, comme c'est le cas du site Web (statique) de la FAQ \LaTeX anglophone, au moteur de recherche Google (et devoir afficher que la recherche est... « *enhanced by Google* »).

FAQ GUTenberg (2020–2023) : Dokuwiki → Sphinx ?

En décembre 2020, j'ai été en mesure de proposer une alternative répondant à ces prérequis : [Sphinx](#). En effet, ce générateur de documentation :

- offre de nombreux formats de sortie :
 - [HTML](#) (site Web statique);
 - [PDF](#) (via \LaTeX)!
 - [EPUB](#), [XML](#), [Texinfo](#), pages « *man* », *plain text*;
- propose comme langage de balisage pour ses sources :
 - [REST](#)³⁵, dans l'esprit voisin de \LaTeX ;
 - [Markdown](#) possiblement, qui est toutefois moins puissant que [REST](#);
 - malheureusement *pas \LaTeX* !
- pour sa sortie [HTML](#) offre (par défaut et gratuitement) :
 - un moteur de recherche *intégré*;
 - un design moderne et réactif;
- est plus qu'éprouvé, avec plus de 15 ans d'existence, et peut être enrichi grâce à une multitude d'extensions;
- est utilisé pour de très nombreuses documentations techniques, dont celle de [Python](#);
- est écrit (et donc personnalisable) en [Python](#).

Néanmoins, cette alternative présentait quelques inconvénients potentiels :

- [Sphinx](#) utilise comme langage de balisage :
 - définitivement *pas \LaTeX* . Toutefois, on peut envisager une conversion depuis et vers \LaTeX via [pandoc](#);
 - [REST](#) dont la syntaxe est au moins aussi horrible et ésotérique que celle de [Dokuwiki](#) !

34. Système de gestion de versions décentralisé.

35. [REST](#) est le langage originel utilisé par [Sphinx](#) et celui qu'il propose avant tout autre.

- **Markdown** qui n'offre pas la finesse de **LAT_EX** ou **REST**. Toutefois, c'est un langage de balisage :
 - simple** : léger, facile à lire et à écrire, comme dans les mails écrits en mode texte ;
 - populaire** : de plus en plus massivement utilisé (et en tout cas bien plus que le langage de balisage de **Dokuwiki**) : fichiers README.md sur le [CTAN](#), [issues](#)³⁶ sur GitHub ou [GitLab](#), messages dans [Discourse](#)³⁷, etc.
- Contrairement à **Dokuwiki**, **Sphinx** n'est pas un outil tout en un. Les fichiers sources de chaque question étant édités indépendamment les uns des autres, il n'y a à priori pas de facilités pour, dans le fichier de l'une d'elles, faire référence à une autre question, voire à une section d'une autre question. On verra ci-dessous que l'éditeur intégré de [GitLab](#) satisfait ce besoin.

Je me suis toutefois lancé dans l'aventure et, en décembre 2020, j'ai travaillé à un prototype, au moins pour une sortie **HTML**. Ce ne fut pas chose aisée du fait de la nécessaire conversion des sources **Dokuwiki** vers **REST**, effectuée par le truchement de **pandoc**, de *custom writers* de ce dernier, de **sed**, de **Perl**, de scripts **bash**, etc. Cette migration fut aussi l'occasion de nombreuses améliorations tant ponctuelles en amont sur le site source **Dokuwiki** (mise en conformité et homogénéisation du code) qu'en masse en aval sur les fichiers **REST**.

Courant février 2021, ce prototype a été soumis au CA de GUTenberg et son accueil par ceux qui se sont exprimés a été, hors Jérémie Just, très favorable. Jérémie Just, quant à lui, n'a donné son avis qu'à une seule reprise : « C'est plutôt chouette ! Le look est plus moderne que celui qu'on a avec notre template Dokuwiki actuel (qui date de 2017) » ; sinon, silence radio : ni approbation, ni désapprobation. J'ai continué à améliorer progressivement ce prototype, par exemple en lui adjoignant un thème moderne, avec mode clair ou sombre, ou en lui greffant **Hypothes.is** qui est un service en ligne gratuit permettant d'annoter, commenter ou surligner n'importe quelle page Web.

Fin janvier 2021, le projet a été présenté aux adhérents lors d'un moment d'échange en ligne et les retours furent peu nombreux mais très favorables. De février 2021 à juin 2023, j'ai dû en arrêter le développement suite à divers points bloquants :

- j'avais l'ambition de pouvoir accompagner les exemples **LAT_EX** minimaux du code (notamment du préambule) nécessaire pour qu'ils soient complets. Ainsi l'exemple montrant comment mettre en couleur du texte :

Exemple 5

```
1 \textcolor{red}{\textbf{Important}}
```

aurait été accompagné, par exemple au moyen d'onglets, du code compilable :

36. Rapports de bogues, demandes de fonctionnalités, etc.

37. Logiciel libre pour forum de discussions intégrant un système de liste de diffusion.

Exemple 6

```

1 \documentclass[french]{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage{lmodern}
5 \usepackage[a4paper]{geometry}
6 \usepackage{xcolor}
7 \usepackage{babel}
8 \begin{document}
9 \textcolor{red}{\textbf{Important}}
10 \end{document}
```

Cela s'est révélé difficile à implémenter et plusieurs pistes explorées n'aboutissaient pas.

- Par ailleurs, la syntaxe `REST`, trop complexe, me paraissait rédhibitoire et à abandonner impérativement au profit de `Markdown`. Cela n'est toutefois pas directement possible et repose sur l'extension tierce `MyST`³⁸ de `Sphinx` qui, pendant plusieurs mois, a été incompatible avec les versions récentes de ce dernier qui offraient certaines fonctionnalités que je jugeais intéressantes, voire indispensables.
- Enfin, j'ai été accaparé par le développement de la classe `LATEX` `letgut` de la *Lettre Gutenberg*.

En juin 2023, ces points bloquants ont été levés :

- L'idée d'accompagner chaque exemple `LATEX` minimal non complet d'une version complète a été abandonnée au profit de blocs de code éditables et compilables³⁹ tels qu'on peut les voir en action sur les sites <https://www.learnlatex.org/fr/> ou `TeXnique`. En particulier, les fragments de code peuvent être complétés automatiquement de façon à devenir des documents minimaux complets !
- L'incompatibilité entre les versions de `MyST` et `Sphinx` a été levée suite à une mise à jour majeure du premier qui a eu en outre l'avantage de rendre les sources en `Markdown` aussi riches en fonctionnalités que celles en `REST` et, dans l'esprit, voisines de `LATEX`.

J'ai donc à nouveau travaillé d'arrache-pied au développement de ce projet (en faisant au passage de nombreux rapports de bugs et demandes de fonctionnalités pour de multiples projets : `pandoc`, `Sphinx`, `MyST`, `TeXLive.net`, etc.) Il m'a notamment fallu me plonger dans le code JavaScript de `TeXLive.net` afin de l'adapter aux particularités de la `FAQ`. Les résultats étaient encourageants : de plus en plus de fonctionnalités intéressantes étaient intégrées et le design du site s'améliorait continuellement.

Néanmoins, plusieurs questions se posaient :

38. `MyST` est un strict sur-ensemble de `CommonMark` dont un des développeurs principaux est John MacFarlane, l'auteur de `pandoc` ; `CommonMark` propose une spécification syntaxique qui règle le problème majeur que présente la syntaxe de `Markdown` de n'être ni standard, ni non ambiguë.

39. Cette fonctionnalité s'appuie sur le serveur `TeXLive.net` qui accepte des documents `LATEX` via une requête `HTTP POST` et renvoie un document `PDF` ou un fichier journal en cas d'erreur. Nous remercions vivement David Carlisle d'avoir développé et, avec Stefan Kottwitz, de fournir à la communauté cet outil extrêmement pratique !

1. celle du flot de travail collaboratif des contributeurs de la FAQ : celui-ci se devait d'être certes efficace, ce qui impliquait de passer par [Git](#), mais également relativement simple, comparable à celui du site sous [Dokuwiki](#). Or la prise en main de [Git](#) est quand même relativement complexe. Toutefois, des plateformes telles que [GitLab](#) peuvent rendre presque transparente la gestion des dépôts sous [Git](#).
2. Celle de l'édition des fichiers sources de la FAQ : les contributeurs qui ne souhaitent pas cloner sur leur machine le dépôt [Git](#) de la FAQ doivent pouvoir éditer ces fichiers de façon simple et, si possible, ergonomique. [GitLab](#) offre depuis longtemps la possibilité, pour chaque fichier individuellement, d'effectuer des modifications via un éditeur en ligne de type [WYSIWYG](#) (prenant en charge la syntaxe [Markdown](#)) et de fournir un message de *commit*⁴⁰ indiquant la nature des modifications (comme c'était le cas sur le site [Dokuwiki](#)). Mais le fait de ne pouvoir modifier qu'un fichier à la fois finit par être peu ergonomique. Toutefois, [GitLab](#) intègre depuis quelques temps un [EDI](#) en ligne permettant d'éditer plusieurs fichiers à la fois. Cet environnement est basé sur (une version limitée de) l'éditeur de code [Visual Studio Code](#), qui est de plus en plus populaire et donc potentiellement déjà utilisé par certains contributeurs ; il offre en outre de nombreuses et très puissantes fonctionnalités :
 - d'édition, notamment en ce qui concerne les fichiers [Markdown](#)⁴¹ ;
 - de recherche dans l'ensemble des fichiers du dépôt ;
 - d'intégration de [Git](#), notamment de gestion des *commits* ;
 - etc.

L'intégration de cet [EDI](#) dans [GitLab](#) présente l'avantage qu'on peut y accéder directement depuis le site Web statique, simplement en cliquant sur une icône, ce qui simplifie énormément le flot de travail des contributeurs qui choisissent de ne pas travailler en local sur leur machine avec un clone du dépôt de la FAQ.

3. Celle de la prise en compte des modifications apportées par les contributeurs qui, sur le site sous [Dokuwiki](#), était instantanée. Or [Sphinx](#) procède de façon similaire à [LATEX](#) : les sources, [REST](#) ou [Markdown](#), doivent être *compilées* pour produire des fichiers « intéressants », notamment [HTML](#)⁴². Afin que les contributeurs puissent rapidement voir l'effet de leurs modifications, il était essentiel que ces compilations puissent être lancées automatiquement, sans intervention manuelle d'un des responsables de la FAQ. Or, justement, [GitLab](#) met à disposition un système [CI/CD](#) permettant cela : les pages [HTML](#) sont régénérées automatiquement à chaque fois que l'on enregistre des modifications sur le dépôt [Git](#) de la FAQ.
4. Celle de la mise en ligne du site Web statique ainsi obtenu : il était là aussi essentiel cela se fasse automatiquement, sans intervention manuelle d'un des responsables de la FAQ. Or, justement, [GitLab](#) met à

40. Dans notre cas, il s'agit en gros de l'enregistrement effectif de modifications apportées à un ou plusieurs fichiers sources de la FAQ.

41. Cf. par exemple <https://code.visualstudio.com/docs/languages/markdown>. On pourra y constater que cet [EDI](#) offre justement des facilités pour, dans le fichier d'une question, faire référence à une autre question, voire à une section d'une autre question (cf. un potentiel [inconvénient](#) signalé ci-dessus).

42. Mais aussi par exemple [PDF](#) via des fichiers [.tex](#).

disposition la fonctionnalité « [Pages](#) » qui permet de publier des sites Web statiques directement à partir d'un dépôt dans [GitLab](#). Ainsi, les modifications apportées aux sources de la FAQ, immédiatement prises en compte grâce au [CI/CD](#) dès qu'elles sont « poussées » sur le dépôt, provoquent automatiquement une nouvelle compilation des pages [HTML](#) concernées et deviennent ainsi rapidement consultables par tout un chacun.

Au début de l'été 2023, les tests que j'ai menés avec ces fonctionnalités de [GitLab](#) montraient des résultats impeccables en terme d'ergonomie. En revanche, en termes de réactivité, j'ai fait face à un problème complètement bloquant ; en effet, la modification d'un unique fichier source [Markdown](#) provoquait :

localemement sur ma machine la régénération quasi-instantanée de l'unique page [HTML](#) correspondante ;
en ligne sur l'instance [GitLab](#) la régénération des 1226 pages [HTML](#) correspondant aux 1226 fichiers sources que comptait alors la FAQ !

En plus d'être tout sauf écologique, cette consommation excessive de ressources durait à chaque fois au moins 15 minutes, ce qui signifiait que le contributeur devait, après chaque modification des fichiers sources « poussées » sur le dépôt [GitLab](#) de la FAQ, attendre tout ce temps pour en voir l'effet sur le site en ligne.

J'ai pensé à priori que ce détail prendrait une journée à être réglé ; j'ai déploré à posteriori que ce détail ait pris tout l'été à être réglé... Mais il a été réglé ! (Pour ceux que cela intéresse, mes péripéties à ce sujet sont détaillées en annexe, page 31.) Dès lors, la modification d'un unique fichier source [Markdown](#) ne provoquait plus que la compilation de l'unique page [HTML](#) correspondante et le délai pour que le site soit actualisé était de l'ordre de la minute.

Un autre point mérite d'être détaillé. Dans la FAQ [Dokuwiki](#), pour que « [LATEX](#) »⁴³ apparaisse bien sous la forme de son logo, le contributeur devait recourir à une syntaxe assez lourde : <[latex](#)>\LaTeX</[latex](#)> ; ensuite, à l'enregistrement du fichier source, un plugin [Dokuwiki](#) lançait à la volée une compilation [LATEX](#) puis en faisait une image [PNG](#) qui, enfin, était intégrée à la page [HTML](#) générée. Pour le lecteur, le résultat était très sous-optimal d'un point de vue typographique, ainsi que le montre la figure 3 : l'image était pixellisée, et la fonte et la famille (serif/sans serif) étaient différentes de celles de l'ensemble du site.

FIGURE 3 – Exemple de logos [TEX](#) et [LATEX](#) sur la FAQ sous [Dokuwiki](#) Vous trouverez ici les réponses aux questions les plus courantes sur [TEX](#), [LATEX](#) et outils associés.

Lors de la migration vers [Sphinx](#), il s'est avéré qu'une telle astuce à base de compilation [LATEX](#) à la volée n'était pas possible. Et, comme j'estimais de toute façon que la méthode basée sur l'inclusion d'images était très sous-optimale, j'ai cherché à faire autrement. J'ai pu mettre au point une solution *en aval*, basée sur un code JavaScript qui, à l'affichage des

43. Ainsi que « [TEX](#) », etc.

pages HTML, remplaçait les occurrences du code « `{latexlogo}`LaTeX`` » (syntaxe restant assez lourde pour le contributeur) par le code du listing 1.

Listing 1

```
<span class="t-logo">
L<span class="a">a</span>T<span class="e">e</span>X
</span>
```

Une classe `css` que j'avais conçue produisait alors un rendu à la fois :

- conforme au véritable logo de Leslie Lamport (ou presque, cf. ci-dessous) ;
- en HTML et pas sous forme d'image ;
- ne jurant pas d'un point de vue typographique (fonte et famille identiques à celles de l'ensemble du site).

FIGURE 4 – Exemple de logos \TeX et \LaTeX sur la FAQ sous Sphinx

Vous trouverez ici les réponses aux questions les plus courantes sur \TeX , \LaTeX et outils associés.

Or, lors du TUG à Bonn mi-juillet 2023, Gerd Neugebauer, connu en particulier pour être le mainteneur du site Web du CTAN, a (bien) voulu regarder comment fonctionnait la FAQ `Sphinx`, notamment du point de vue du contributeur. Je ne sais plus pourquoi, la page HTML que j'ai choisie au hasard affichait « `LaTeX` » et non « `\LaTeX` », et il en a fait le reproche. Je lui ai montré qu'il s'agissait là d'un accident, l'ensemble du site affichant bien le logo ; ensuite, par simple clic sur une icône de la page Web, nous avons abouti au fichier source de cette page et Gerd a vu par quelle syntaxe cela passait pour le contributeur (« `{latexlogo}`LaTeX`` », donc), ce qu'il m'a signalé comme étant inutilement lourd. En effet, pour le site du CTAN, il a pour cela une solution *en amont* : avant leur publication, les fichiers HTML sont tous scannés par un script Perl qui y remplace les occurrences de « `LaTeX` » (seulement) par le code HTML du listing 1 ; ensuite, là aussi grâce à un fichier CSS de son cru, le rendu est conforme au véritable logo de Leslie Lamport. Gerd m'a très aimablement fourni son script et son fichier CSS, ce qui fait que, désormais, les contributeurs de la FAQ `Sphinx` n'ont plus qu'à écrire « `LaTeX` » dans les fichiers sources pour que, dans les pages HTML résultantes, ce soit bien « `\LaTeX` » qui apparaisse ! En outre, dans la classe CSS que j'avais conçue, mes choix de positions des lettres « A » et « E » étaient selon moi élégants mais n'étaient en fait pas tout à fait conformes au logo de Leslie Lamport ; en revanche, avec le fichier CSS de Gerd, le rendu est conforme. Nous remercions vivement Gerd pour tout cela et pour avoir été à l'écoute des retours que Bastien Dumont (qui a récemment rejoint de CA de GUTenberg) et moi avons faits pour signaler de légers bugs et proposer des pistes visant à rendre plus puissante encore sa solution⁴⁴.

44. C'est désormais un fork, créé par Bastien, qui est en œuvre sur le site de la FAQ.

FAQ GUTenberg (2023–aujourd’hui) : Dokuwiki → Sphinx !

En septembre 2023, le CA de GUTenberg a considéré que le projet était suffisamment mature pour que la FAQ [Dokuwiki](#) soit migrée vers [Sphinx](#). Néanmoins, nous souhaitions réduire encore le temps de compilation et pour cela, il semblait nécessaire (et suffisant) que nous disposions de notre propre instance [GitLab](#) ainsi que de notre propre [GitLab runner](#)⁴⁵. Ceci nécessitait une montée en puissance de notre serveur, ce qui a été acté par le CA.

Maxime Chupin a alors réalisé un considérable travail d’administration système pour installer, via des images [Docker](#), nos propres instances de [GitLab](#)⁴⁶ et [GitLab runner](#), mais aussi pour trouver une alternative « manuelle » à [Pages](#) que nous ne parvenions pas à faire fonctionner sur notre instance; et ça n’aura pas été un travail inutile puisque, désormais, le temps mis au processus pour actualiser le site après modification d’un unique fichier source [Markdown](#) est passé d’un peu plus d’une minute à environ 30 s. Maxime a également testé la FAQ du point de vue du contributeur et en a créé un nouveau logo.

FIGURE 5 – Exemple de page de la FAQ sous [Sphinx](#) (vue ordinateur)

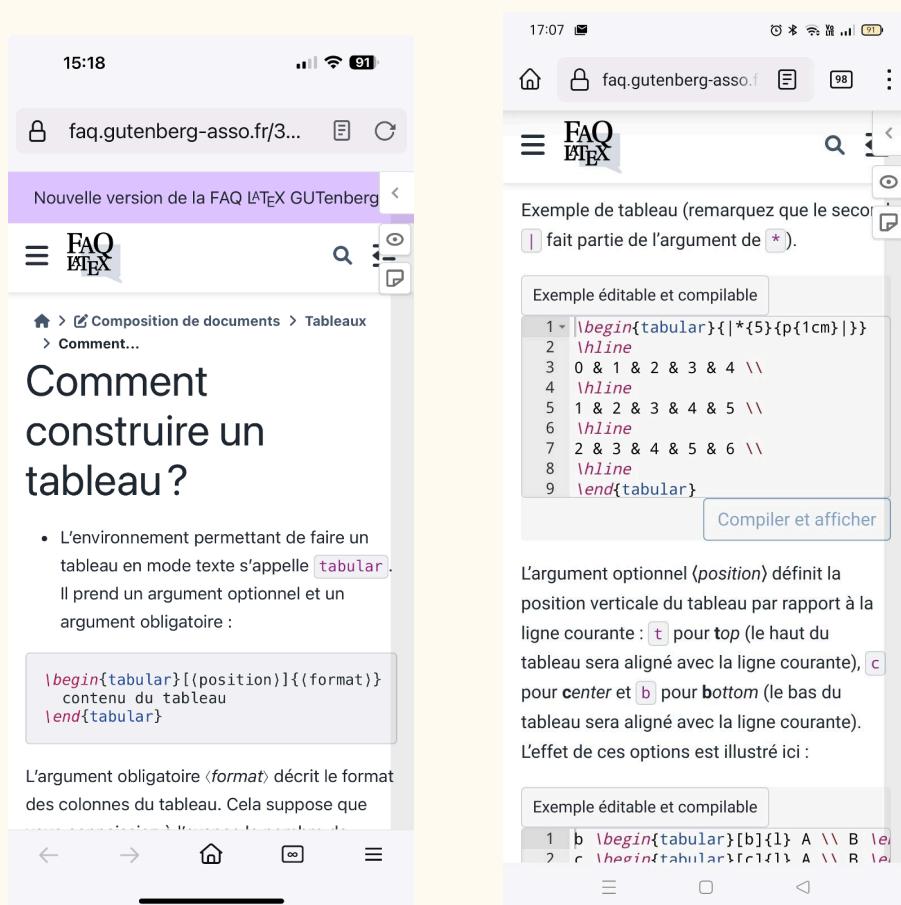
Maxime et moi avons aussi cherché à élucider un mystère... En effet, début 2023, Jérémy Just a mis en ligne un *fork* de la FAQ GUTenberg et ce, sans l’annoncer au CA de l’association, ni avant, ni après l’avoir fait. Il va sans dire que nous déplorons au plus haut point cet état de fait car il dessert considérablement la communauté des utilisateurs francophones de L^AT_EX : ceux qui cherchent de l’aide se retrouvent désormais avec deux FAQ divergeant progressivement! Mais ce qui reste mystérieux, c’est que

45. Le *runner* est un logiciel, connecté à l’instance [GitLab](#), qui se met en écoute des demandes de tâches, notamment donc de la compilation de la FAQ lorsqu’un *commit* est « poussé » sur le dépôt.

46. Cette instance se trouve à l’adresse <https://gitlab.gutenberg-asso.fr> et a pour but d’héberger aussi les dépôts des productions francophones autour de (L)A_TE_X.

depuis début 2023 et jusqu'à récemment (donc que ce soit avant ou après la migration vers [Sphinx](#)), les moteurs de recherche, notamment Google, n'ont plus *du tout* répertorié la FAQ GUTenberg (qui n'a pas changé d'[URL](#) entre temps); et seule celle de Jérémie Just était répertoriée, et très bien répertoriée⁴⁷ ! Même si nous avons notre idée sur la question, on peut se demander quelle est l'explication de ce mystère.

FIGURE 6 – Exemple de page de la FAQ sous Sphinx (vue téléphone portable)



Finalement, après de nombreux ajustements complémentaires, la migration de la FAQ de [Dokuwiki](#) vers [Sphinx](#) a eu lieu fin novembre 2023. Elle peut être trouvée à l'adresse suivante :

<https://faq.gutenberg-asso.fr>

mais voici des captures d'écran, à comparer avec les précédentes, montrant le gain en terme de design et d'ergonomie.

47. Heureusement, grâce à Maxime qui s'est plongé dans *Google Search Console*, la FAQ GUTenberg est désormais (depuis début 2024 environ) répertoriée, et régulièrement mieux que celle de Jérémie Just.

Plusieurs contributeurs ont rejoint cette nouvelle version de la FAQ GUTenberg :

- certains anciens : Maxime Chupin, moi-même, Bastien Dumont et Yannick Tanguy (toujours aussi extraordinairement actif!);
- mais aussi des nouveaux : Patrick Bideault, Yann Denichou, Bernard Peyréga et Wissam Sabra.

Un grand merci à eux tous!

FAQ GUTenberg (aujourd’hui – futur)

Nous espérons que cette nouvelle version Sphinx de la FAQ, avec toutes les fonctionnalités qu'elle apporte⁴⁸, rendra service à la communauté francophone des utilisateurs de (L^A)T_EX.

Nous espérons également qu'elle suscitera des vocations de contributeur⁴⁹ : cher lecteur, c'est avec plaisir que la joyeuse équipe des contributeurs actuels vous accueillera en son sein!

Pour le futur, il est évident que nous allons continuellement améliorer, corriger, actualiser et développer la FAQ. Parmi les pistes possibles de développement, il serait intéressant et certainement très utile est de concevoir une intelligence artificielle basée sur le trésor que représente le très riche contenu de cette FAQ.

Annexe : détails de mes péripéties

Je livre ici les détails des péripéties évoquées ci-dessus.

1. Je me suis donc battu de début juillet à mi-août 2023, parfois pendant des journées entières, pour régler un problème de la nouvelle FAQ extrêmement bloquant : la modification d'un unique fichier source provoquait la régénération, non pas seulement de l'unique page HTML correspondante, mais des 1226 pages HTML des 1226 fichiers sources ! J'ai remué ciel et terre, posé la question sur un nombre incalculable de forums, ai sollicité Maxime Chupin, Paul Gaborit, Georges Khaznadjar, ai passé des heures à voir si ça marcherait mieux sur une instance auto-hébergée de [GitLab](#) avec un *runner* auto-hébergé (ça ne marchait pas mieux mais j'ai réussi à créer une telle instance), etc. Je suis passé par des moments de doute terribles. Et puis, finalement, je suis tombé tout à fait par hasard et, surtout, par miracle (cf. ci-dessous) sur une solution, extrêmement simple, qui a fait passer la régénération, après modification d'un unique fichier source, d'environ 15 minutes à un peu plus d'une minute.
2. Ce qui est tout à fait ahurissant dans cette histoire, c'est que je suis donc tombé sur la solution ci-dessus, non seulement tout à fait par hasard, mais aussi par pur miracle. En effet, la machinerie ci/cd décrite ci-dessus nécessite un système d'exploitation et un certain nombre de programmes qui, pour des questions de sécurité, de reproductibilité,

48. Déttaillées dans l'article « Fonctionnement de la FAQ L^AT_EX GUTenberg » page 33.

49. Cf. l'article « Fonctionnement de la FAQ L^AT_EX GUTenberg », section « Améliorations » page 46.

d’automatisation, etc., sont rendus disponibles dans des environnements virtualisés isolés via des « images », par exemple des images *Docker*. Pendant des semaines, l’image *Docker* que j’ai utilisée était une brique immuable constituée d’une distribution Linux minimale agrémentée d’une distribution *Python* (nécessaire puisque *Sphinx* est codé en *Python*) sur laquelle, à chaque modification de la *FAQ*, étaient ensuite installés un certain nombre d’outils dont notamment *Sphinx*, l’ensemble lançant enfin la compilation *Sphinx* des fichiers sources. Je me disais depuis longtemps qu’il faudrait que je parte plutôt d’une image *Docker* intégrant déjà *Sphinx* : Paul Gaborit et Georges Khaznadar l’ont fait pour moi début juillet 2023 et ça montrait une légère accélération du processus de démarrage, mais rien au niveau de la compilation ; j’avais cessé d’utiliser ces images car elles utilisaient la version 6.2.1 de *Sphinx* alors que, à partir de mi-juillet, je pouvais (et devais pour des raisons de compatibilité) utiliser la version la plus récente (7.1.2) de *Sphinx*. Lorsque, en désespoir de cause, j’ai décidé mi-août d’à nouveau tester une image *Docker* intégrant *Sphinx*, je ne pouvais me baser sur celle officiellement fournie par *Sphinx* car elle était basée sur une version de *Sphinx* antérieure à la 7.1.2 ; en butinant sur Internet, j’ai trouvé une autre image *Docker* intégrant *Sphinx* 7.1.2 qui, elle, a comme par magie réglé le problème de compilation interminable ! J’ai signalé la chose aux développeurs de *Sphinx* qui quelques jours plus tard ont fourni une nouvelle image *Docker* officielle intégrant *Sphinx* 7.1.2 ; je l’ai testée mais, pour une raison toujours pas claire, le problème de compilation interminable persistait avec cette image officielle. La solution trouvée relève donc du miracle : j’aurais pu tomber sur mille autres images *Docker* intégrant *Sphinx* 7.1.2 avec lesquelles le problème aurait persisté !

Références

- [1] Bobby BODENHEIMER. « *TEX, LATEX, etc.* questions et réponses ». fr. Trad. par Philippe LOUARN et al. In : *Cahiers GUTenberg* 13 (1992), p. 55-77. URL : http://www.numdam.org/item/CG_1992___13_55_0/.
- [2] GUTENBERG’S OFFICE. « FAQ : questions souvent posées sur (L)ATEX ». fr. In : *Cahiers GUTenberg* 23 (1996), p. 1-114. URL : http://www.numdam.org/item/CG_1996___23_1_0/.
- [3] Marie-Paule KLUTH. *FAQ LATEX française. Pour débutants et confirmés*. Paris : Vuibert, 1999. 440 p. ISBN : 2-7117-8641-2.
- [4] Marie-Paule KLUTH. *FAQ LATEX. Foire aux questions LATEX*. 2^e éd. Paris : Vuibert, 2000. 441 p. ISBN : 2-7117-8662-5.

Denis Bitouzé



FONCTIONNEMENT DE LA FAQ LATEX GUTENBERG

Résumé

L'objet de cet article est d'indiquer comment fonctionne la version actuelle de la FAQ LATEX GUTenberg du point de vue utilisateur.

Plan de l'article

Page d'accueil	33
Page point d'entrée de thématique	36
Page standard	37
Fonctionnalités diverses	41
Système d'annotations	43
Déficiences et améliorations de la FAQ	44

La FAQ LATEX GUTenberg est accessible à l'utilisateur à la page :

<https://faq.gutenberg-asso.fr>

La navigation y est intuitive. Nous vous invitons à feuilleter les pages de la FAQ ou à y chercher des informations en cliquant en haut à droite sur la petite loupe Q.

Nous détaillons ci-dessous les fonctionnalités de la FAQ, mais il n'est nul besoin de les connaître toutes pour commencer à l'utiliser !

Page d'accueil

Examinons tout d'abord la page d'accueil de la FAQ, dont un aperçu se trouve figure 7.

FIGURE 7 – Page d'accueil de la FAQ

The screenshot shows the homepage of the LaTeX Gutenberg FAQ. At the top, there's a purple header bar with the text "Nouvelle version de la FAQ LATEX GUTenberg !". Below the header, the main content area has a light blue background. On the left, a large text box says "Bienvenue sur la FAQ LATEX francophone" and "Vous trouverez ici les réponses aux questions les plus courantes sur TeX, LATEX et outils associés." It features a "Par où commencer ?" button. To the right of this box is a large "FAQ LATEX" logo. Further right, there's a sidebar with links like "Sur cette page", "Foire aux questions/Frequently asked questions (FAQ)", "D'autres ressources", "Ça ne marche pas !", and "Que faire pour aider ?". At the bottom, there are several cards for different topics: "Généralités", "Programmation", "Composition de documents", "Domaines spécialisés", "Fichiers utilisés", and "Logiciels et distributions". Each card contains a brief description of its content.

On y voit immédiatement que les questions/réponses de la FAQ sont classées par les grandes thématiques suivantes :

- « Généralités »
- « Programmation »
- « Composition de documents »
- « Domaines spécialisés »
- « Fichiers »
- « Logiciels »
- « Cette FAQ »
- « Contribuer »

Ces thématiques sont listées à la fois dans la zone en haut de (chaque) page et dans des vignettes qui les détaillent succinctement. Dans les deux cas, ce sont des hyperliens vers les pages « points d'entrée » (cf. ci-après) de ces thématiques. Une vignette supplémentaire « Une page au hasard ? » permet de consulter une page choisie aléatoirement⁵⁰.

FIGURE 8 – Recherche dans la FAQ

The screenshot shows the LaTeX GUTenberg FAQ search interface. At the top, there are two search results for the query "couleur":

- Composition de documents**: A detailed page about document composition, including tables, figures, and annexes.
- couleur**: A page specifically about color usage in LaTeX, including sections on changing page colors and defining new colors.

Below these, the main search results for "couleur" are displayed:

- Recherche**: A search bar with the query "couleur".
- Résultats de la recherche**: A list of 46 results found, including:
 - Cellules**: Information on how to merge cells in a table.
 - Changer la couleur de fond des pages**: Instructions on using the xcolor package to change page backgrounds.
 - Comment définir de nouvelles couleurs ?**: Information on defining new colors.
 - Comment écrire du texte en couleurs ?**: Information on writing colored text.
 - Paragraphe**: Information on setting paragraph colors.
 - Comment barrer ou biffer du texte ?**: Information on striking through text.

Dans la zone visible en haut de chaque page de la FAQ se trouvent également :

- (à gauche) , le logo de la FAQ qui est un lien hypertexte ramenant à la page d'accueil;
- (à droite) quatre icônes groupées :

50. Et, le cas échéant, d'y contribuer ; cf. section « Améliorations » page 46.

1. **Q** qui, lorsqu'elle est cliquée, ouvre une boîte de saisie permettant d'effectuer des recherches dans la FAQ, comme l'illustre la figure 8 page précédente. On peut aussi à tout instant faire apparaître cette boîte de saisie au moyen du raccourci clavier **Ctrl**+**K**.
2. **☀, ☺** ou **🌙** indiquant le mode respectivement clair, sombre ou automatique⁵¹ du thème du site et permettant d'en changer. Pour plus de clarté, c'est avec le thème clair que les copies d'écran ont été faites ;
3. **🔗** qui est un lien vers le dépôt Git de la FAQ sur l'instance **GitLab** de GUTenberg ;
4. **gut** qui est un lien vers le site de l'association GUTenberg.

Pour finir sur ce qu'on observe tout en haut de la page, notons tout à droite, en haut de la barre de défilement vertical de la page, les trois boutons **<**, **⌚** et **↗**. Ceux-ci seront discutés section « **Système d'annotations** », page 43.

Ensuite, dans le bandeau sur fond coloré se trouve un bouton « **Par où commencer ?** » qui est un lien interne vers une page donnant quelques pistes sur la façon dont on peut aborder la lecture de la FAQ. On remarque que, lorsqu'on passe le pointeur de la souris sur ce lien interne, et comme sur tous ceux de la FAQ, une info-bulle contenant le début de la page en lien apparaît (cf. figure 9)⁵².

Le reste de la page est du texte (contenant d'autres liens internes) structuré en sections dont on peut remarquer, dans le volet de droite, la table des matières locale « **☰ Sur cette page** » (dont chaque entrée est un lien vers la section correspondante). Dans ce volet, se trouvent également deux liens que nous ne détaillons pas ici : « **📝 Modifier (Ctrl + ➡ conseillé)** » et « **📄 Montrer le code source** » utiles respectivement aux contributeurs (cf. section « **Améliorations** » page 46) et aux personnes intriguées par la présente FAQ. Ce volet se retrouve sur chaque page de la FAQ et contient une table des matières locale si (et seulement si) le contenu est structuré en sections.

FIGURE 9 – Liens internes et info-bulles



51. Qui suivra la valeur par défaut du système qui a été définie.
52. C'est également le cas des hyperliens vers les pages Wikipédia anglophones et DOI. Nous avons bon espoir de pouvoir bientôt étendre cette fonctionnalité aux liens vers les pages Wikipédia (au moins) francophones et vers les pages CTAN des classes et packages (L)ATEX.

Page point d'entrée de thématique

Si on clique par exemple sur la thématique « Composition de documents » (soit via le lien en haut de page, soit via la vignette correspondante), on aboutit à la page illustrée figure 10.

FIGURE 10 – Page point d'entrée de la thématique « Composition de documents »

The screenshot shows a web browser displaying a page titled 'Composition de documents'. The page is part of the 'FAQ LaTeX GUTENBERG !' website. The navigation bar at the top includes links for 'Généralités', 'Programmation', 'Composition de documents' (which is the active tab), 'Domaines spécialisés', 'Fichiers', 'Logiciels', 'Cette FAQ', and 'Contribuer'. Below the navigation bar, there is a sidebar titled 'Navigation de la section' containing links for 'Annexes', 'Documents', 'Flottants', 'Illustrations', 'Langues', 'Tableaux', and 'Texte'. The main content area contains a heading 'Composition de documents' with a sub-headline: 'Ce thème regroupe toutes les questions traitant de la composition de documents courants, pouvant inclure des tables, des figures ou des annexes. Voici dans le détail les sujets qu'elle regroupe :'. It lists several topics: 'les différents documents et classes associées', 'la composition du texte', 'la gestion des spécificités des différentes langues', 'les annexes', 'les flottants', 'les tableaux', and 'les illustrations'. At the bottom of the page, there are links for 'Précédent' (Comment écrire des scripts interactifs ?) and 'Suivant' (Annexes).

Plusieurs choses sont à y noter :

- Le texte détaille encore davantage la thématique, avec des hyperliens vers ses sous-thématiques et, le cas échéant, des sous-sous-thématiques.
- Comme sur toutes les pages, des liens situés tout en bas permettent de passer aux questions précédentes et suivantes.

FIGURE 11 – Navigation de la section en cours dans le volet gauche

The screenshot shows a sidebar titled 'Navigation de la section' on the left side of the page. This sidebar lists various document components: 'Annexes', 'Documents', 'Création d'un document', 'Inclusion de fichier dans un document', 'Flottants', 'Illustrations', 'Langues', 'Tableaux', and 'Texte'. Each item has a dropdown arrow to its right. Under 'Création d'un document', there are three links: 'Comment découper un document en plusieurs fichiers ?', 'Comment inclure du code Plain T_EX dans L_AT_EX ?', and 'Pourquoi L_AT_EX ne veut pas inclure mon fichier ?'. Under 'Inclusion de fichier dans un document', there are two links: 'Que fait la commande \include ?' and 'Comment inclure des fichiers sans modifier leurs liens internes ?'. Under 'Flottants', there is one link: 'Métadonnées d'un document'. Under 'Version de document', there is one link: 'Flottants'. Under 'Illustrations', there is one link: 'Langues'. Under 'Langues', there is one link: 'Tableaux'. Under 'Tableaux', there is one link: 'Texte'.

- Comme sur toutes les pages (autres que celle d'accueil) se trouve un volet de gauche :
 - affichant un aperçu des sous-thématiques de la thématique en cours ;
 - donnant le moyen d'y naviguer : il s'agit d'un arbre pouvant être déployé en cliquant sur ses « branches » et dont les « feuilles » sont des questions, hyperliens vers les pages correspondantes.

Cf. figure 11 page ci-contre.

Page standard

Visitons désormais une page standard. Pour cela, j'invite le lecteur à lancer dans la FAQ une recherche (icône **Q** ou raccourci clavier **Ctrl**+**K**), comme ci-dessus sur le mot « couleur », puis de cliquer sur la question « [Comment écrire du texte en couleurs ?](#) ». La première chose qu'on constate alors est que les occurrences du mot « couleur » y sont surlignées et que, comme c'est généralement souhaité, ce surlignage peut être supprimé en cliquant sur le bouton « Cacher les résultats de la recherche ». La page résultante est celle affichée en figure 12.

FIGURE 12 – Page standard

The screenshot shows a web browser displaying the LaTeX Gutenberg FAQ. The URL in the address bar is "Nouvelle version de la FAQ LaTeX Gutenberg !". The page title is "Composition de documents > Paragraphes > Comment...". The main content area contains the question "Comment écrire du texte en couleurs ?" followed by a detailed answer about the \textcolor command. On the left, there is a sidebar titled "Navigation de la section" with a tree view of the document structure. At the bottom right, there is an "Attention" box stating that the \textcolor package is no longer used.

On voit alors d'abord un hyperlien précédé de l'icône ; celle-ci identifie dans l'ensemble de la FAQ tout hyperlien vers la page CTAN d'une classe ou d'un package. D'autres types d'hyperliens spécifiques émaillent la FAQ :

- ceux vers la documentation en ligne sur <https://texdoc.net> d'une classe ou d'un package, identifiés par l'icône
- ceux vers une page de la FAQ anglophone <https://texfaq.org/>, identifiés par l'icône
- ceux vers une page Wikipédia (francophone ou pas), identifiés par l'icône
- ceux vers la page d'un livre sur la base de données ISBN <https://isbndb.com/>, identifiés par l'icône

On trouve ensuite un bloc de code, ici **LATEX**, contenant :

Exemple 7

```
1 \textcolor{<couleur>}{{texte à mettre en couleur}} % portée locale
2   \color{<couleur>} % portée semi-globale (bascule)
```

Il est à noter que ce code peut être copié très simplement : en passant le pointeur de la souris sur le bloc, un bouton y apparaît en haut à droite et, s'il est cliqué, l'ensemble du code en question est copié dans le presse-papiers.

Le texte qui suit est parsemé d'extraits de code en ligne (c'est-à-dire intégrés aux paragraphes) qui sont mis en évidence, par exemple `<couleur>`.

On note ensuite l'exemple **LATEX** :

Exemple 8

```
1 \documentclass[french]{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage{lmodern}
5 \usepackage[a4paper]{geometry}
6 \usepackage{xcolor}           % À ne pas oublier !
7 \usepackage{babel}
8 \begin{document}
9 \textcolor{red}{Important !}
10 Pas important.
11 {\color{red} Important !}
12 Pas important.
13 \end{document}
```

dont il est annoncé qu'il est... *compilable et éditable*! Et, en effet, cet exemple (ainsi que de nombreux autres dans la FAQ) est :

compilable : en cliquant sur le bouton « Compiler et afficher » (ou automatiquement à l'ouverture de la page⁵³) le code est compilé en ligne⁵⁴ puis, si la compilation :

- aboutit, le **PDF** résultant est affiché en dessous, dans une fenêtre offrant les fonctionnalités les plus courantes des afficheurs **PDF**, comme l'illustre la figure 13 page suivante.
- échoue, le fichier `.log` est affiché à la place du **PDF**.

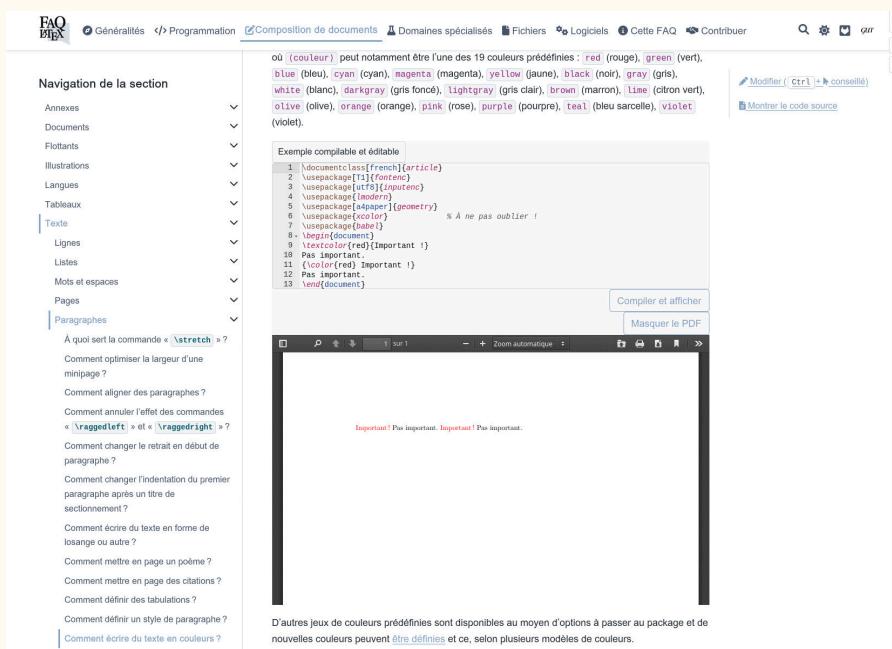
Au besoin, on peut masquer la sortie au moyen du bouton « Masquer le **PDF** ».

éditable : l'utilisateur peut à sa guise apporter des modifications au contenu puis en visualiser l'effet en relançant la compilation au moyen du bouton « Compiler et afficher » (cf. figure 14 page 40).

Il est à noter que, une fois qu'il a le focus, ce code peut lui aussi être très simplement copié dans le presse-papiers, notamment au moyen des raccourcis clavier usuels **Ctrl**+**A** (tout sélectionner) puis **Ctrl**+**C** (copier).

53. Nous n'avons pas encore bien compris pourquoi cet automatisme n'est pas systématique.
 54. Grâce à [TeXLive.net](#), cf. article « [Petite histoire de la FAQ LATEX GUTenberg](#) » page 20.

FIGURE 13 – Page standard (suite) : exemple compilé et affiché



Comme sur de nombreuses pages de la FAQ, on trouve plus bas dans cette page certains passages mis en évidence au moyen de ce qu'on appelle des « admonitions » (cf. figure 14 page suivante).

Celles-ci sont les suivantes :

- « **!** Attention »;
- « **▲** Prudence »;
- « **▲** Danger »;
- « **✖** Erreur »;
- « **?** Indication »;
- « **!** Important »;
- « **i** Note »;
- « **↗** Voir aussi »;
- « **?** Astuce »;
- « **⚠** Avertissement ».

Comme sur la plupart des pages, les sources ayant servi à établir les réponses à la question en cours figurent en fin de page, dans une admonition « **i** Sources » spécifique.

Enfin, on voit tout en bas de la page la date de dernière mise à jour de la question/réponses en cours.

Ajoutons toutefois un élément qui a pu échapper à la vigilance du lecteur : tout en haut de chaque page, immédiatement sous la zone constamment visible, se trouve un fil d'Ariane (éventuellement tronqué selon la profondeur à laquelle est la page et selon la longueur des titres qui y sont affichés). L'icône située tout à gauche permet de revenir « à la maison », c'est-à-dire à la page d'accueil.

FIGURE 14 – Page standard (suite) : exemple modifié puis compilé et affiché

The screenshot shows a LaTeX editor interface. On the left is a sidebar with a tree view of sections like Annexes, Documents, Flottants, Illustrations, Langues, Tableaux, Texte, Lignes, Listes, Mots et espaces, Pages, and Paragraphes. The 'Texte' section is expanded. In the main area, there's a code editor with the following LaTeX code:

```

1 \documentclass[french]{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[utf8]{inputenc}
4 \usepackage{lmodern}
5 \usepackage{color}
6 \usepackage{geometry} % A ne pas oublier !
7 \usepackage{babel}
8 \begin{document}
9 Coucou, je suis en \textcolor{blue}{couleur} !
10 Pas moi.
11 Coucou, je suis en \color{orange} couleur !
12 Pas moi.
13 \end{document}

```

Below the code editor is a preview window showing the text "Coucou, je suis en couleur ! Pas moi. Coucou, je suis en couleur ! Pas moi." and a "Compiler et afficher" button. To the right of the preview is a sidebar with a "Composition de documents" tab selected, showing a list of color names: red, green, ver, blue, cyan, magenta, yellow, black, gray, white, blanc, darkgray, lightgray, brown, marron, lime, citron vent, olive, olive, orange, pink, purple, teal, bleu sarcelle, violet.

FIGURE 15 – Page standard (suite) : admonitions, sources, etc.

This screenshot shows a standard LaTeX documentation page. The sidebar on the left is identical to Figure 14. The main content area includes an "Exemple compilable et éditable" box with the same LaTeX code as Figure 14. Below it is an "Attention" box stating: "Le package \color, utilisé historiquement, est de nos jours supplanté par \xcolor." Further down are "Voir aussi" and "Sources" sections. The "Voir aussi" section links to "Au sujet des couleurs" and "Pour colorer l'arrière-plan des paragraphes". The "Sources" section links to "Documentation des packages \xcolor et \color". At the bottom, there are navigation links for "Précédent" and "Suivant", and a footer noting the page was last updated on March 21, 2024, at 20:59.

Fonctionnalités diverses

Nous listons ici un peu pêle-mêle diverses fonctionnalités offertes par la FAQ.

Onglets

Certains passages de la FAQ sont répartis dans différents onglets, notamment lorsqu'il est opportun de comparer les sorties PDF de codes L^AT_EX voisins. C'est par exemple le cas à la question « [Comment améliorer la qualité typographique de son document ?](#) » où l'on peut visualiser l'effet du package `microtype` (cf. figures 16 et 17 de la présente page et page suivante).

FIGURE 16 – Comparaison au moyen d'onglets (1)

Avant même l'amélioration esthétique globale que `\usepackage{microtype}` apporte, il a généralement un effet spectaculaire, en réduisant les coupures de mots nécessaires en fin de ligne, et les débordements dans la marge. Voici un exemple. À droite, nous avons juste ajouté la ligne `\usepackage{microtype}` :

The screenshot shows a LaTeX editor interface. At the top, there are two tabs: "Sans `microtype`" and "Avec `microtype`". Below them is a code editor with the following content:

```

1 \documentclass[french]{article}
2 \usepackage[T1]{fontenc}
3 \usepackage[paperwidth=5.5cm,paperheight=6cm,margin=0.5cm]{geometry}
4 \usepackage{lmodern}
5 \usepackage{babel}
6 % \usepackage{microtype}
7 \pagestyle{empty}
8 \begin{document}
9 Longtemps, je me suis couché de bonne heure. Parfois, à peine ma bougie éteinte,
10 mes yeux se fermaient si vite que je n'avais pas le temps de me dire : \og{}Je
11 m'endors.\fg{} Et, une demi-heure après, la pensée qu'il était temps de chercher
12 le sommeil m'éveillait ; je voulais poser le volume que je croyais avoir encore
13 dans les mains et souffler ma lumière.
14 \end{document}

```

To the right of the code editor are two buttons: "Compiler et afficher" (Compile and display) and "Masquer le PDF" (Hide PDF). Below the code editor is a preview window showing the resulting PDF document. The PDF contains a single paragraph of text:

Longtemps, je me suis couché de bonne heure. Parfois, à peine ma bougie éteinte, mes yeux se fermaient si vite que je n'avais pas le temps de me dire : « Je m'endors. » Et, une demi-heure après, la pensée qu'il était temps de chercher le sommeil m'éveillait ; je voulais poser le volume que je croyais avoir encore dans les mains et souffler ma lumière.

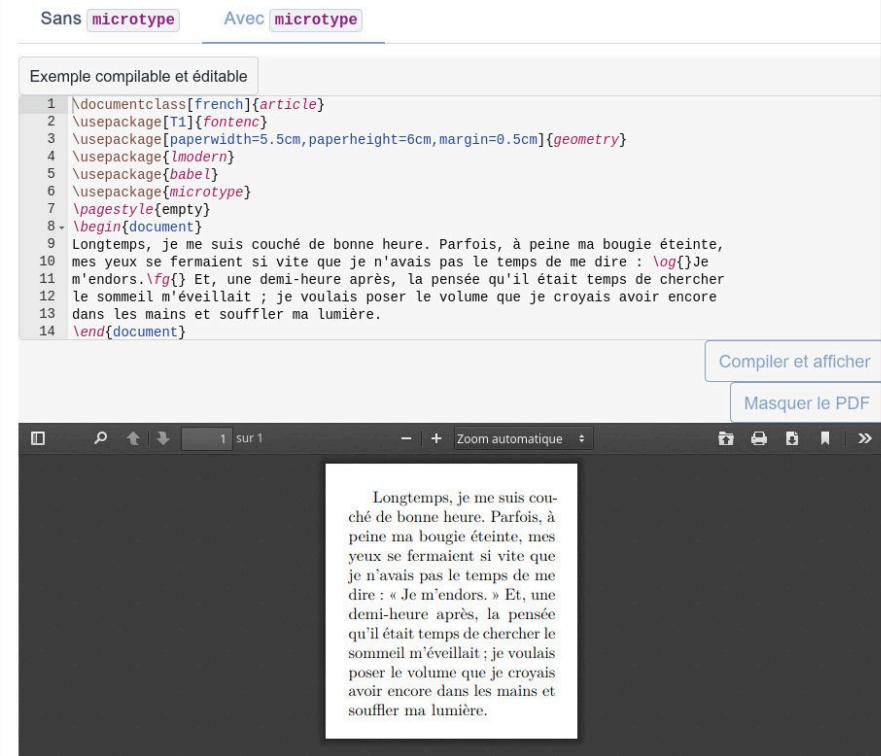
Exemples de codes automatiquement complétés

Le système assurant la compilation en ligne de blocs de codes L^AT_EX, [TeXLive.net](#), dispose d'une fonctionnalité ingénieuse, même si elle est peu visible pour le visiteur : certains blocs de code *compilables et éditables* ne contiennent au départ que l'extrait « intéressant » du corps du document et sont, à l'ouverture de la page, automatiquement et presque instantanément complétés (de façon heuristique) en des exemples complets minimaux⁵⁵ !

55. Donc avec la commande `\documentclass`, un préambule minimal et englobant l'extrait dans l'environnement `document`.

FIGURE 17 – Comparaison au moyen d'onglets (2)

Avant même l'amélioration esthétique globale que `\microtype` apporte, il a généralement un effet spectaculaire, en réduisant les coupures de mots nécessaires en fin de ligne, et les débordements dans la marge. Voici un exemple. À droite, nous avons juste ajouté la ligne `\usepackage{microtype}` :



Ainsi, à l'ouverture de la page « [À quoi sert la commande \mbox ?](#) », le premier exemple de code **LATEX**, qui est *compilable et éditable*, ne contient que :

Exemple 9

```

1 \section{Avant}
2 Pour votre information, mon numéro de téléphone est désormais le
3 01 02 03 04 05. Il remplace mon ancien numéro de téléphone
4 qui ne doit maintenant plus être utilisé.
5
6 \section{Après}
7 Pour votre information, mon numéro de téléphone est désormais le
8 \mbox{01 02 03 04 05}. Il remplace mon ancien numéro de téléphone
9 qui ne doit maintenant plus être utilisé.

```

et, en l'état, provoquerait une erreur de compilation ; mais il est immédiatement complété en :

Exemple 10

```

1 % Début du code ajouté automatiquement
2 \documentclass[french]{article}
3 \usepackage[T1]{fontenc}
4 \usepackage[utf8]{inputenc}
5 \usepackage{lmodern}
6 \usepackage[a4paper]{geometry}
7 \usepackage{babel}

```

```

8
9   \begin{document}
10  % Fin du code ajouté automatiquement
11
12  \section{Avant}
13  Pour votre information, mon numéro de téléphone est désormais le
14  01 02 03 04 05. Il remplace mon ancien numéro de téléphone
15  qui ne doit maintenant plus être utilisé.
16
17  \section{Après}
18  Pour votre information, mon numéro de téléphone est désormais le
19  \mbox{01 02 03 04 05}. Il remplace mon ancien numéro de téléphone
20  qui ne doit maintenant plus être utilisé.
21
22  % Début du code ajouté automatiquement
23  \end{document}
24  % Fin du code ajouté automatiquement

```

Le système fait même mieux : en examinant les commandes utilisées dans l'exemple, il ajoute au préambule le chargement des packages non standard éventuellement nécessaires, au moins pour certaines des commandes de certains packages.

Système d'annotations

Un système d'annotations est fourni avec la FAQ. Il repose sur le service [Hypothesis](#) gratuit et ne nécessitant de compte que pour être auteur de notes.

Pour accéder à ce service disponible sur chaque page, il suffit de cliquer sur les boutons propres à Hypothesis situés tout en haut de la barre de défilement vertical :

- < qui affiche (et > qui masque) le volet latéral propre à Hypothesis. Celui-ci contient d'éventuelles notes attachées à la page qui ont été publiées⁵⁶ :
 - soit, publiquement, par d'autres personnes ;
 - soit, publiquement ou pas, par le visiteur.

Les notes publiques ont notamment pour vocation de partager avec les autres visiteurs de la FAQ des informations (précisions, erreurs, etc.) relatives à la page en cours ; à ce sujet, cf. section « [Améliorations](#) », page [46](#).

Les notes peuvent faire l'objet :

- de commentaires (bouton ↲), publics ou privés, ajoutés par les personnes ayant un compte Hypothesis ;
- de recherche (bouton ⌂) ou de classement (bouton ↑), si elles sont nombreuses ;
- de partage par [URL](#) (bouton ↗), voire d'exportation ou d'importation.
- ⓟ qui masque (et ⓘ qui affiche) les passages ayant été surlignés (cf. ci-après) et éventuellement annotés, soit par le visiteur, soit (publiquement) par d'autres personnes. Les passages surlignés sont

⁵⁶. Il est nécessaire d'avoir un compte Hypothesis pour pouvoir attacher des notes aux pages de la FAQ.

numérotés et repérables verticalement dans la page au moyen de leurs numéros figurant dans des flèches (cliquables) positionnées sur le bord gauche du volet Hypothesis.⁵⁷

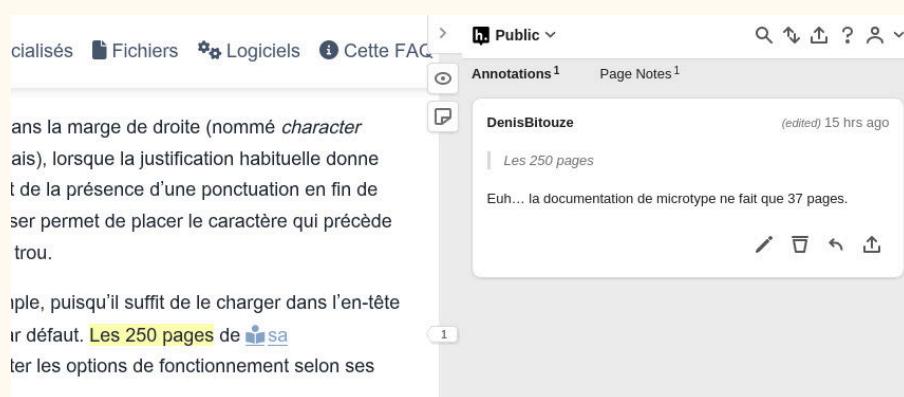
- (resp. si un passage est sélectionné, cf. ci-après) qui affiche le volet Hypothesis directement dans l'onglet dédié à l'ajout d'une nouvelle note de la page (resp. d'une nouvelle annotation de passage surligné) publique ou privée (compte Hypothesis requis).

En outre, dès lors que le visiteur sélectionne un passage d'une page, un menu contextuel lui permet, s'il possède un compte Hypothesis, de surligner seulement (bouton !) ou d'en plus annoter (bouton !) le passage en question.

Le visiteur ayant un compte Hypothesis peut y retrouver, centralisés, les notes et passages surlignés qu'il a attachés à la FAQ (et à tout autre site proposant le même service).

Un exemple de ces fonctionnalités se trouve à la page « [Comment améliorer la qualité typographique de son document ?](#) » et est illustré figure 18. On voit

FIGURE 18 – Vue du volet Hypothesis



que le volet Hypothesis comporte deux onglets, « *Annotations* » et « *Pages Notes* », qui contiennent ici respectivement un⁵⁸ passage publiquement surligné et annoté (« Les 250 pages... ») et une⁵⁸ note publiquement publiée.

Déficiences et améliorations de la FAQ

Déficiences

La FAQ n'est pas parfaite et ne le sera jamais. Voici quelques-uns de ses aspects méritant améliorations.

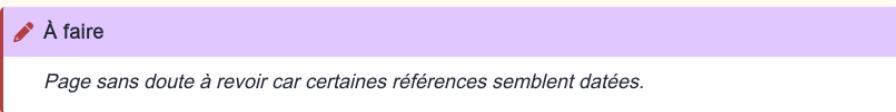
Admonitions « À faire ». Les contributeurs remarquent parfois qu'une réponse à une question nécessite d'être actualisée, révisée ou approfondie (voire supprimée), ou bien ne sont pas sûrs de certains aspects de la réponse qu'ils sont être en train de rédiger. Dans ces cas-là, ils le

57. Notons qu'il est possible, au moyen d'un clic gauche, de naviguer entre un passage surligné dans la page de la FAQ et l'entrée correspondante dans le volet Hypothesis. En outre, lorsqu'on clique sur un passage déjà surligné dans la page de la FAQ, un filtre (désactivable) masque dans ce volet les notes ainsi que les entrées des passages surlignés non encore cliqués de la page.

58. Et un(e) seul à l'heure où ces lignes sont écrites.

signalent à l'attention des visiteurs et des autres contributeurs de la FAQ dans une admonition « À faire ». C'est par exemple le cas au début de la question « [Quels sont les éditeurs utilisables sous macOS ?](#) », comme l'illustre la figure 19.

FIGURE 19 – « À faire »
dans la FAQ



Ces « À faire » sont listés à la page « [Quelles réponses sont à réviser ?](#) ». Bien entendu, il y a inévitablement des pages nécessitant d'être améliorées mais pas signalées par des « À faire » ; à ce sujet, cf. section « [Améliorations](#) », page suivante.

Hyperliens obsolètes. Vu l'âge de la FAQ, il n'est pas étonnant que certains hyperliens qu'elle contient soient obsolètes. Ils sont alors signalés ainsi :

Mark van Dongen, [LATEX and friends](#)^{lien obsolète}. Ce site présente le livre de l'auteur mais aussi un ensemble de présentations détaillant par thèmes les fonctionnalités de LATEX.

et sont répertoriés [sur notre instance GitLab](#).

Pages en anglais. Ainsi que l'explique l'article « [Petite histoire de la FAQ LATEX GUTenberg](#) » page 20, la FAQ francophone s'est très tôt partiellement nourrie des questions de la FAQ anglophone en les intégrant telles quelles. Même si un grand nombre ont été traduites, il en reste certaines devant encore l'être. Elles sont répertoriées à la page « [Quelles pages traduire ?](#) ».

Pages d'accueil de sections. Les pages qui accueillent le visiteur quand il arrive dans une section ou une sous-section ont été par défaut automatiquement générées par [Sphinx](#) lors de la migration depuis [Dokuwiki](#), contenant alors une simple liste des sous-sections et des pages qui se trouvent à la racine de la section courante. Un grand nombre de ces pages ont été révisées de façon à ce que leur contenu explique quel est l'objet de la section et de ses sous-sections, et pour exposer la matière dans un ordre plus pertinent. Il en reste cependant plusieurs n'ayant pas encore été remaniées. On pourra consulter la page « [Comment améliorer les pages start.html ?](#) » pour voir comment les choses peuvent être améliorées.

Passages en multicolonne. Dans la FAQ [Dokuwiki](#) dont est issue la présente FAQ, certaines pages contenaient des passages en multicolonne, souvent pour comparer les sorties PDF de codes LATEX voisins. Lors de la migration vers [Sphinx](#), ces présentations en parallèle n'ont pas pu être conservées. Aussi peut-être certaines des pages concernées, listées à la page « [Pages originellement avec passages en multicolonne](#) », devraient-elles reprendre la présentation originale, au moins dans l'esprit (par exemple au moyen d'onglets : cf. section « [Onglets](#) » page 41).

Pages avec codes en échec. Certaines pages de la FAQ recèlent des codes LATEX (complets) dont la compilation échoue. Répertoriées à la page

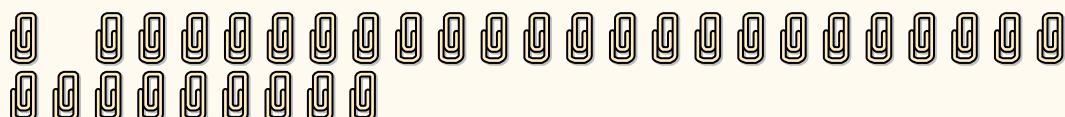
« Pages avec codes \LaTeX (complets) en échec », elles méritent que ces codes soient révisés.

Améliorations

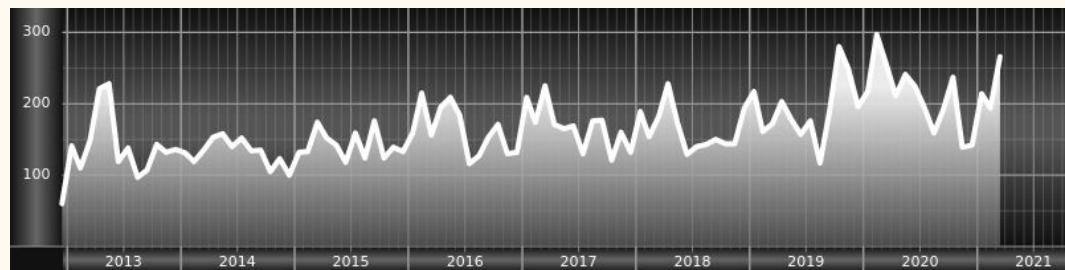
Il va sans dire que nous invitons les visiteurs de la FAQ à nous aider à l'améliorer :

- soit en signalant des déficiences et en proposant des pistes d'amélioration, par le biais de la liste dédiée faq@gutenberg-asso.fr ou du système d'annotations [Hypothesis](#);
- soit, mieux encore, en devenant contributeur ! Cela fera probablement l'objet d'un article futur, mais les personnes intéressées peuvent d'ores et déjà consulter la page «  [Contribuer](#) » pour avoir un aperçu du fonctionnement de la FAQ du point de vue du contributeur. Et l'exposé du 2 mai sera consacré à la FAQ : voir page 18. Nous rappelons, cher lecteur, que c'est avec plaisir que la joyeuse équipe des contributeurs actuels vous accueillera en son sein !

Denis Bitouzé



ET MAINTENANT, UNE BONNE VIEILLE VEILLE TEXNOLOGIQUE !



Chers adhérents, nous veillons $\text{\TeX}nologiquement$ pour vous !

En effet, la présente rubrique est dédiée aux nouveautés apparues sur le CTAN que vous auriez pu manquer. Elle témoigne de la vitalité de la communauté \TeX . 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.

Enfin, nous avons à cœur d'illustrer ces pages par des exemples. La plupart sont dûs aux auteurs des packages eux-mêmes : nous les avons trouvés dans leurs documentations et nous en publions le code en regard du résultat. Mais ce code est parfois trop long pour être publié en ces pages, auquel

cas seul le résultat est utilisé; il est néanmoins facile à trouver dans la documentation du package en question.

Nouveautés

Pour la rubrique du présent numéro, nous listons la plupart des nouveautés, classées par ordre chronologique, apparues depuis la précédente *Lettre*⁵⁹ et jusqu'au mois de mars 2024 inclus^{60, 61}. Nous espérons n'oublier aucun nouveau package. Si c'était le cas, merci de nous le faire savoir.

Novembre 2023

autotype : a pour ambition d'adapter certains mécanismes de \TeX aux normes spécifiques à certaines langues. Pour le moment, seul l'allemand est supporté. Le lexique de l'allemand comprend en particulier de nombreux mots composés, comme *Standardtrennverfahren* (« comportement standard de coupure de mots »), composé de *standard*, *Trenn* (coupure) et *Verfahren* (comportement) : *autotype* implémente la pondération des points de coupure pour que \TeX ait plus de chances de couper, par exemple, après « trenn » que après « stand » ou « ver ». De même, il fait en sorte qu'un mot composé comme *Auffahrt* ne présente pas de ligature entre ses deux « f ». Pareils ajustements aux convention d'une langue particulière sont généralement assurés par des interventions directes de l'utilisateur, quand ils le sont ; les recherches et expérimentations conduites par les auteurs pour l'automatiser autant que possible devraient intéresser au-delà des cercles germanophones. Ce package a fait l'objet d'un article détaillé dans la dernière livraison de *Die \TeXnische Komödie*. Son usage y est illustré par des exemples. [1]

○ **coloredbelts** : permet d'insérer, dans un document \LaTeX , un pictogramme représentant une ceinture colorée.

quran-id : fournit quelques macros pour composer tout ou partie du Coran. Ce package fournit aussi quelques commandes pour des traductions en allemand, anglais et perse.

notebeamer : est une classe qui permet de présenter un diaporama beamer sous forme de « cahiers de notes » (*notebook*) avec, à gauche, les diapositives alignées verticalement et, à droite, un espace avec des lignes pour prendre des notes.

quran-en : est une extension du package *quran-id* pour la langue anglaise.

litetable : est une classe de document, construite à partir de la classe *standalone*, qui fournit de nombreuses commandes pour présenter le déroulement temporel de cours (plutôt orientés universitaires).

argumentation : est un package (dont les macros peuvent être interprétées par *TikZ*) permettant de produire des diagrammes illustrant

59. La précédente *Lettre* était numérotée 51; comme vous l'avez lu dans la note 5 de l'éditorial, en page 2, elle est parue le 12 novembre 2023.

60. Nous avons arrêté le contenu de cette *Lettre* début avril; le délai séparant ce moment de la parution s'explique par les relectures et corrections, qui ont bien occupé la rédaction et les correcteurs.

61. Nous présentons nos excuses aux auteurs qui ont publié des packages depuis début avril. Ces derniers seront dûment chroniqués dans la *Lettre* 53.

des raisonnements logiques dans le champs d'étude de la théorie de l'argumentation.

q-and-a : est un package permettant de mettre en page des conversations du type *questions-réponses* à partir d'un texte simple ressemblant à du MarkDown. Cela fait écho aux *prompts* utilisés pour interagir avec les intelligences artificielles génératives.

affilauthor : est un package permettant de remplir les informations (nom, adresse électronique, téléphone, URL, etc.) des auteurs et des affiliations sous forme de champs « clé/valeur » de la commande \authos et \affil.

polyhedra : est un package, donc la documentation n'est hélas pas terminée, qui semble fournir des commandes pour représenter des polygones et des polyèdres.

○ **ipsum** : est un package, qui comme **lipsum**, permet simplement de produire du texte de remplissage dans différentes langues (dont le français). Il prend le contenu fournit par le site web <https://ipsum.one/>.

Exemple 11

1 \ipsum<Lang=FR>[2]

code

résultat

On rit beaucoup, on admira ; tel est l'esprit à l'usage de Verrières. Julien était déjà debout, tout le monde se leva malgré le décorum ; tel est l'empire du génie. Mme Valenod le retint encore un quart d'heure ; il fallait bien qu'il entendît les enfants réciter leur catéchisme ; ils firent les plus drôles de confusions, dont lui seul s'aperçut. Il n'eut garde de les relever. Quelle ignorance des premiers principes de la religion ! pensait-il. Il saluait enfin et croyait pouvoir s'échapper ; mais il fallut essuyer une fable de La Fontaine.

Sur ce, rappelons que, pour mieux prêter attention à l'empagement, il vaut mieux utiliser du texte de remplissage dans une langue que l'on ne comprend guère. Ainsi, le *vulgum pecus* choisira le latin, comme en atteste, au hasard, le fil suivant, issu d'un site propulsé par l'association : <https://texnique.fr/osqa/questions/9333/koma-script-e-t-police-de-numero-de-page>, tandis que le plus insigne des latinistes choisira un langue qu'il ne maîtrise pas (non sans avoir corrigé le barbarisme précédent).

pmdraw : permet de faciliter la création de diagrammes de monoïdes^{62, 63}.

gitinfo-lua : un package pour LualATEX permettant d'afficher les informations d'un dépôt Git dans un document PDF. Ce package n'est utilisable qu'avec lualatex et l'option --shell-escape.

○ **sim-os-menus** : est un package pour simuler des fenêtres de visionnage, des terminaux et des menus contextuels de systèmes d'exploitation standards.

62. [https://fr.wikipedia.org/wiki/Monoïde_\(théorie_des_catégories\)](https://fr.wikipedia.org/wiki/Monoïde_(théorie_des_catégories))

63. En cas de problèmes matutinaux d'avance à dépression sur un delco, le laboratoire de neurophysiologie de la Faculté de médecine de l'hôpital Saint-Antoine, à Paris, recommande l'ingestion immédiate de deux monoïdes. Ils règlent le problème dans 87,4 % des cas. [2]

Exemple 12

code
résultat

```

1 \ContextMenu[Font=\sffamily,ListSeps={1,3/1/4/1/},
2   ListOffsets={1,2,1,0}]{%
3   Menu 1(>),Menu 2,Menu 3(>)(*),Menu 4 §
4   Sous-menu A(>),Sous-menu B(>)(* §
5   Sous-sous-menu 1,Sous-sous-menu 2,Sous-sous-menu
6     3(>),Sous-sous-menu
7     4(*),Sous-sous-menu 5(>)
}

```

litesolution : est une classe au design sobre et élégant, dédiée à la composition d'examens et de leurs solutions.

latex2pydata : permet de transférer des données depuis L^AT_EX vers Python en utilisant la syntaxe littérale de Python.

cid : renommée **cidarticle** peu après son apparition, cette classe est dédiée à la revue *Commentarii informaticae didacticae*, publiée par l'université de Potsdam, en Allemagne. Découverte à l'occasion de la rédaction de cette notule, cette revue s'avère passionnante, à commencer par sa sixième livraison, consacrée à l'enseignement de l'informatique à l'école⁶⁴, que nous recommandons chaudement à nos lecteurs anglophones.

➊ **TrivialPursuit** : est un package permettant de représenter un plateau de jeu du style du plateau du célèbre *Trivial Pursuit* tout en permettant de personnaliser les couleurs ainsi que les logos (qui proviennent du package **fontawesome5**).

Exemple 13

code

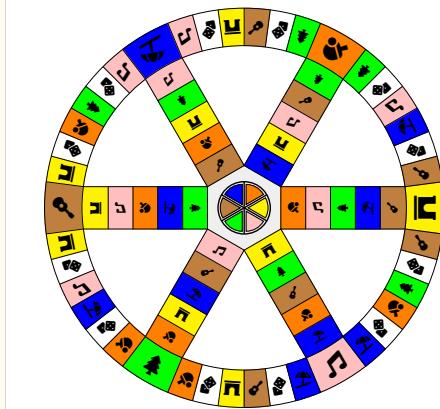
```

1 \PlateauTrivialPursuit[Unite=0.33]

```

⁶⁴. On trouvera tous les numéros de la revue à l'adresse <https://publishup.uni-potsdam.de/opus4-ubp/solrsearch/index/search/searchtype/series/id/9/start/0/rows/10>.

résultat (suite)



pdfannotations : est un utile package qui fournit des commandes et des environnements pour annoter des PDF et plus particulièrement des diapositives PDF.

texblend : est un programme permettant, à l'aide de commentaires particuliers, de compiler des sous-parties d'un document principal qui les inclut par des commandes \input.

- **vectorlogos** : fournit des logos au format vectoriel pour les logiciels emacs, geogebra, scratch, texmaker et texstudio.

Décembre 2023

- **cjs-rcs-article** : est la classe de *La Revue canadienne de statistique*, en anglais le *Canadian Journal of Statistics*. Notons que cette revue est publiée par la société « Canards Illimités Canada », sise à Stonewall, dans l'État du Manitoba (le nom anglais de cette société étant tout naturellement *Ducks Unlimited Canada*).
- heria** : est une classe facilitant le dépôt de demandes de financement auprès de la commission européenne, via son programme *Horizon Europe*⁶⁵, et plus précisément pour les programmes *Research and Innovation Action (RIA* – recherche et action en matière d'innovation) et *Innovation Action (IA* – action en faveur de l'innovation).
- pdfrender** : permet d'ajuster finement l'empagement ainsi que d'autres paramètres affectant l'apparence des PDF. Ce package de Heiko Oberdiek⁶⁶ faisait auparavant partie du package **oberdiek**. Il est à présent indépendant.
- **tutodoc** : est utilisé par son auteur pour produire de façon sémantique des documentations de packages et de classes L^AT_EX dans un style de type tutoriel, et avec un rendu sobre pour une lecture sur écran.
- **decimalcomma** : permet d'utiliser la virgule comme séparateur décimal. En effet, en mode mathématique, la virgule est toujours, par défaut, considérée comme un symbole de ponctuation. Le package **decimalcomma** permet, comme **icomma** et **nccomma**, de gérer ce problème (y compris pour d'autres langues que le français), avec moins d'incompatibilité, notamment avec L^AT_EX ou X_EL^AT_EX.

65. Horizon Europe est le principal programme de financement de l'UE pour la recherche et l'innovation. https://commission.europa.eu/funding-tenders/find-funding/e-u-funding-programmes/horizon-europe_en?prefLang=fr&etrans=fr

66. On trouve les nombreux packages de cet auteur sur sa page CTAN : <https://ctan.org/author/oberdiek>

iran-bibtex : regroupe les styles BibTeX conformes aux lignes directrices énoncées au sein du manuel de style de l'Iran. [3] Cette classe de Farshad Rasuli dispose d'une documentation en farsi doublée d'une assez complète version anglaise, accessible via un fichier `readme`.

responsive : est un package permettant de produire des documents dans différentes dimensions pour les différents formats de page, les liseuses, et même les téléphones. Un des principaux éléments gérés par le package est le corps des fontes, qui est modifié selon les dimensions du document final.

- ◉ **thematicpuzzle** : est un utile package qui permet de créer un bandeau horizontal, sous forme de puzzle, avec différentes thématiques illustrées par des logos du package `fontawesome5`.

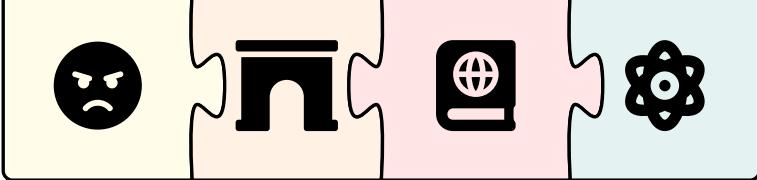
Exemple 14

code

```

1 \PuzzleThemes%
2 [Echelle=2.5,CouleursFond={yellow!10,orange!10,red
  !10,teal!10},
3 Labels={Maths,Histoire,Géo,Chimie}]%
4 {\faAngry,\faArchway,\faAtlas,\faAtom}
```

résultat



Maths Histoire Géo Chimie

xkeymask : est une extension du package `xkeyval` qui fournit des macros pour masquer certaines clés pour les rendre inutilisables. Pour en savoir plus, mieux vaut commencer par comprendre ce que fait `xkeyval`!

pgfkeysearch : fournit des commandes permettant d'obtenir les valeurs de clés PGF.

Janvier 2024

amnestyreport : est une classe pour les rapports d'Amnesty International.

freealign : fournit des commandes pour aligner des formules mathématiques sur différentes lignes (en mode mathématique *in line*).

junicodevf : est un package facilitant l'utilisation de la fonte *Junicode*, une police de caractères Unicode libre (sous la SIL Open Font License). Junicode a été dessinée pour être utilisée par des médiévistes et fait partie de la Medieval Unicode Font Initiative (MUFI).

tikzquests : est un package fournissant des commandes permettant de composer des examens (suite de questions et solutions) à partir de répertoires d'exercices à paramètres. Cependant, à la différence de packages comme `exam`, `exam-n` ou `exam-lite`, `tikzquests` n'impose pas de style ni de forme pour la composition.

- tikzdotncross** : fournit des commandes TikZ permettant de styliser les chemins (path) qui se coupent (à la manière des conventions dans la représentation des schémas électriques).
- verifycommand** : ce package fournit des commandes permettant de vérifier si la définition de certaines commandes ou environnements n'a pas été changée. Cela permet en particulier, pour les auteurs de packages, de vérifier cela avant de *patcher* une commande.
- medmath** : vient corriger quelques défauts de l'option mediummath du package **nccmath** pour fournir des commandes mathématiques dédiées aux éléments de taille moyenne (comme les opérateurs de somme ou d'intégrale) qui sont plus robustes et flexibles.
- lua-placeholders** : est un package Lual^AT_EX permettant d'utiliser **YAML** pour gérer les paramètres d'un document.
- tikzquads** : fournit, pour TikZ, quelques formes de boîtes complémentaires pour être utilisées avec **CircuiTikz**.
- regulatory** : est un package pour les professionnels du droit au sens large. Il fournit des commandes pour les références croisées, et utilise notamment **BIB**T_EX et **glossaries**.
- odesandpdes** : fournit des commandes pour les opérateurs différentiels pour les équations aux dérivées ordinaires et les équations aux dérivées partielles.
- ppmcheck** : est un package de conversion de fichiers **PDF** en **PNG** et de validation. Il est devenu **ppmcheckpdf** sur T_EX Live mais reste appelé **ppmcheck** sur MiK^TE_X.
- latex-dependency-grapher** : est un programme java, en ligne de commande, permettant de visualiser les dépendances d'un fichier L^AT_EX. Il crée un graphique avec GraphViz pour les visualiser.
- beamerthemeconcrete** : est une collection de thèmes **beamer**.
- randexam** : est une classe L^AT_EX permettant de mettre en page des sujets d'examens et de tirer aléatoirement certains paramètres des énoncés. Il est principalement conçu pour les examens de mathématiques.
- moloch** : est un thème **beamer**. Il s'agit d'un *fork* du thème **metropolis**.
- genealogy-profiles** : est un package destiné à la généalogie, qui permet de mettre en forme des fiches de profil à partir d'informations renseignées sous forme de clé/valeur.
- undar-digitacion** : permet d'illustrer les doigtés de notes de musique pour la flûte à bec, la quena, le pinquillo et le saxophone.

Février 2024

- Ysabeau** : fournit le support de la fonte Ysabeau pour les moteurs pdfT_EX et dvips. Il ne fournit hélas pas la configuration **fontspec** pour Lual^AT_EX ou X_EL^AT_EX.
- udepcolor** : est un package L^AT_EX qui fournit les couleurs de l'université de Piura⁶⁷, au Pérou, en utilisant **xcolor**.
- emotion** : permet de composer des émoticones à partir de leur caractères Unicode directement saisis en argument de la commande \emotion.

67. https://fr.wikipedia.org/wiki/Université_de_Piura

contract : est un package facilitant, pour les avocats et les notaires, la composition de contrats, de statuts et de commentaires légaux.

tblr-extras : fournit deux extensions au package `tabulararray`⁶⁸ : la première modifie les légendes de `talltblr` et `longtblr` pour utiliser les réglages du package `caption`, et la deuxième redéfinit `conthead-text` et `contfoot-text` de l'environnement `longtblr` pour pouvoir utiliser le langage défini par `babel`.

- **pictochrono** : est un package fournissant une commande pour dessiner un pictogramme de chronomètre.

Exemple 15

code

```

1 \Large Vous n'avez désormais plus que
2 \pictochrono[Delta=15,Color=blue,ColTime=red]{20}
   minutes pour finir la relecture
3 de cette \lettre{}\footnote{C'est ce que les
   relecteurs de la présente
4 livraison ont pensé pendant tout le mois d'avril
   .}.
```

résultat

Vous n'avez désormais plus que  minutes pour finir la relecture de cette *Lettre*^a.

a. C'est ce que les relecteurs de la présente livraison ont pensé pendant tout le mois d'avril.

- **pynotebook** : est un package permettant de reproduire un *Jupyter notebook*⁶⁹ soit en utilisant simplement `listings` sans exécution de code Python, soit en utilisant `piton` et `pyluatex` avec `LuaLaTeX` et `--shell-escape`.

Exemple 16

code

```

1 \begin{NotebookMarkdown}[flush right]{0.9
  \ linewidth}
2 {\Large\bfseries Ceci est un test de bloc \textsf{
  Markdown}.}\ \\
3 Il est possible d'utiliser des formules \LaTeX{},%
  comme ceci : %
4 [
5 \left\{\begin{array}{l}
6 F_0 = 0 \\
7 F_1 = 1 \\
8 F_{n+2} = F_{n+1} + F_n
\end{array}\right.
```

68. À propos de `tabulararray`, nous vous renvoyons au passionnant exposé que Paul Gaborit lui consacra le 4 avril. Il sera bientôt en ligne. <https://www.gutenberg-asso.fr/4-avril-2024-Expose-sur-le-package-tabulararray>

69. Un *Jupyter notebook* n'a rien à voir avec le jupitérisme cher aux débuts présidentiels de M. Macron : ledit *Jupyter* est une application web de programmation. https://fr.wikipedia.org/wiki/Jupyter#Jupyter_Notebook

```

9   \end{array}\right. .
10  \]
11  \end{NotebookMarkdown}
12  \begin{NotebookIn}*[flush right]{0.9\linewidth}
13  def fibonacci_aux(n,a,b):
14      if n == 0 :
15          return a
16      elif n == 1 :
17          return b
18      else:
19          return fibonacci_aux(n-1,b,a+b)
20
21  def fibonacci_of(n):
22      return fibonacci_aux(n,0,1)
23
24  [fibonacci_of(n) for n in range(10)]
25  \end{NotebookIn}
26  \begin{NotebookOut}*[flush right]{0.9\linewidth}
27  [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
28  \end{NotebookOut}

```

code (suite)

Ceci est un test de bloc Markdown.

Il est possible d'utiliser des formules L^AT_EX, comme ceci :

$$\begin{cases} F_0 = 0 \\ F_1 = 1 \\ F_{n+2} = F_{n+1} + F_n \end{cases}$$

```
In [ ]: 1 def fibonacci_aux(n,a,b):
2     if n == 0 :
3         return a
4     elif n == 1 :
5         return b
6     else:
7         return fibonacci_aux(n-1,b,a+b)
8
9 def fibonacci_of(n):
10    return fibonacci_aux(n,0,1)
11
12 [fibonacci_of(n) for n in range(10)]
```

```
Out[ ]: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

résultat

sjtutex : permet de composer des documents au sein de l'université Jiao-tong de Shanghai⁷⁰.

typstfun : est un document PDF qui liste les équivalents entre L^AT_EX et les fonctions Typst⁷¹.

twoxtwogame : fournit des outils pour représenter, visualiser, et nommer

70. https://fr.wikipedia.org/wiki/Université_Jiao-tong_de_Shanghai

71. Typst est un système de composition de documents relativement récent. <https://typst.app>

des jeux sous forme normale à deux joueurs (en théorie des jeux et en économie).

cleveref-forward : permet, lors de la déclaration d'un label, de faire apparaître les références à ce label. Par exemple, il permet de faire apparaître le théorème dans lequel une équation est utilisée.

beautynote : est une élégante classe dédiée à la composition de livres de cours.

weiqi : est dédié au jeu de go. Cette classe dispose d'une documentation rédigée en chinois. Nous déplorons de ne pas maîtriser cette langue ; nous serions très heureux que quelque sinophone rejoigne la rédaction. Dans l'attente, nous nous plongeons dans la lecture d'un utile traité de go, dont nous recommandons particulièrement la section 0.6. [4]

didec : permet de composer des nombres décimaux avec deux décimales (comme cela se fait dans nombre de transactions financières, et ce dans des devises diverses⁷²).

Mars 2024



chemformula-ru : permet l'utilisation du package **chemformula** lorsque babel est configurée pour la langue russe.

useclass : permet de charger des classes LATEX en tant que package.

fontscale : permet de modifier simplement les tailles de fontes d'un document (notamment des commandes classiques \tiny, \scriptsize, etc.). Et ce, localement ou globalement !

fadingimage : permet, en utilisant la bibliothèque fading de TikZ, d'insérer en fond de page une image qui s'estompe petit à petit. Un exemple illustré figure ci-contre.

litebook : est une classe d'usage aisé, destinée à la composition de livres, et notamment de support de cours.

chicagolinks : est un style de bibliographie. Il fait référence au code typographique *The Chicago Manual of Style*⁷³, très utilisé aux États-Unis d'Amérique.

I3sys-query : est, à la manière du programme Java **texosquery**, un programme Lua permettant d'obtenir des informations (moins limitées que ce qu'on peut obtenir avec T_EX seul) de façon plus sécurisée. Pour l'instant, il est utilisable en programme externe mais sera, dans le futur, accessible depuis **expl3**.

○ **lato-math** : fournit la fonte **OTF** mathématique *Lato Math*, conçue pour être utilisée avec les moteurs **LuaT_EX** ou **X_ET_EX** conjointement avec les fontes *Lato*, de Łukasz Dziedzic, mises à notre disposition par Mohamed El Morabity et son package **Lato**.

Exemple 17

code

```

1  \setmathfont{Lato Math}
2  \setmainfont{Lato}
3  \setsansfont{Lato}[Extension = .ttf,
```

72. « Allitérez, allitérez, disait le rhétoricien. Il en restera toujours quelque chose. »

73. https://fr.wikipedia.org/wiki/The_Chicago_Manual_of_Style

code (suite)

```

4   UprightFont = *-Regular, BoldFont = *-Bold,
5   ItalicFont = *-Italic,
6   BoldItalicFont = *-BoldItalic]
7   \sffamily
8   Et une équation :
9   \[ \frac{\partial f}{\partial x} = \alpha \mathbf{V} + \boldsymbol{\nabla} \Gamma + \beta \mathbf{M}
10  + \boldsymbol{\beta} \mathbf{M} \]

```

résultat

Et une équation :

$$\frac{\partial f}{\partial x} = \alpha \mathbf{V} + \boldsymbol{\nabla} \Gamma + \beta \mathbf{M}$$

dashrulex : fournit des commandes pour tracer diverses lignes pointillées dans le texte courant d'un document.

ideavault : permet de créer des concepts avec des informations, liés ou inclus dans d'autres concepts, et d'y affecter des *tags*. Cela fait, le package permet d'afficher ces concepts et leurs descriptions, soit individuellement, soit partiellement ou même par lots, ceux-ci étant classés par tags. Ce package aidera ceux qui ont des idées et désirent les coucher sur le papier en utilisant l'outil informatique.

sfee : est une classe pour le magazine *Smart Factory and Energy Efficiency*⁷⁴.

iaria : est une classe non officielle pour les publications de l'IARIA⁷⁵.

iaria-lite : est une version allégée de la classe **iaria** : elle ne fournit pas la configuration **biblatex**, ceci pour permettre l'utilisation de **BIBTEX**.

thmlist : est un package fournissant des commandes pour définir simplement des environnements de type théorème. Ce package utilise **amsthm** et **mdframed**.

cs-techrep : est une classe pour composer des rapports techniques en sciences informatiques et dans l'ingénierie logicielle, en respectant des conventions bibliographiques de l'IEEE⁷⁶.

longmath : est un package LualATeX qui fournit diverses commandes permettant de gérer les formules sur plusieurs lignes avec des délimiteurs appairés (tels des parenthèses⁷⁷). Ce package exploite les possibilités offertes par Lua.

repthoerem : permet de définir des théorèmes (ou autres environnements similaires) et de pouvoir les répéter simplement et y faire référence.

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

74. <https://sfee.space>

75. L'IARIA est l'*International Academy, Research, and Industry Association*. <https://www.aria.org/>

76. https://fr.wikipedia.org/wiki/Institute_of_Electrical_and_Electronics_Engineers

77. CQFD.

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.

Patrick Bideault, Maxime Chupin & Bastien Dumont

Références

- [1] Keno WEHR. « Sprachspezifische Typographie mit autotype ». In : *Die TExnische Komödie* 36.1 (fév. 2024), p. 23-44.
- [2] Georges PEREC. « Experimental demonstration of the tomatotopic organization in the Soprano (*Cantatrix Sopranica L.*) » In : *Banana Split* n° 2 (1980).
- [3] *Iran Manual of Style*. Iranian Research Institute for Information Science and Technology (IranDoc). 2016. URL : <https://en.irandoc.ac.ir/service-product/1641>.
- [4] Pierre Lusson, Georges PEREC et Jacques ROUBAUD. *Petit Traité invitant à la découverte de l'art subtil du go*. Christian Bourgois éditeur, 1969.



LUA^{TEX} ET METAPOST AU SERVICE DE LA VULGARISATION

Récemment, j'ai participé à l'accueil d'élèves de classe de troisième dans mon laboratoire de recherche en mathématiques. J'ai souhaité les introduire à une partie de mon métier : le calcul scientifique⁷⁸. Nous avons donc, avec un collègue, décidé de construire un exposé autour du calcul numérique de π . Vous pouvez d'ailleurs consulter le support de la présentation à l'adresse :

<https://www.ceremade.dauphine.fr/~chupin/exposes/2024/methodesNum.pdf>

Ici, je voudrais illustrer comment **Lua^{TEX}** et **MetaPost** m'ont permis de produire les supports graphiques de nos explications.

Lua^{TEX} et MetaPost avec `minim-mp`

Comme on peut le constater dans mes divers articles pour la *Lettre*, je suis un fervent utilisateur de **MetaPost**. Son intégration au moteur **Lua^{TEX}** me facilite encore son utilisation. J'ai parlé de nombreuses fois du package **luamplib** qui fournit l'environnement **mplibcode** permettant d'exécuter du code **MetaPost** dans le source **L^AT_EX**.

Dans cet article, je vais montrer que l'on peut utiliser un autre package : **minim-mp**. Celui-ci fait partie du format **minim plain** pour le moteur **Lua^{TEX}**, format qui implémente le support de nombreuses fonctionnalités de **Lua^{TEX}** et du format **PDF**. La plupart d'entre elles sont fournies au travers de packages séparés qui peuvent être utilisés séparément :

minim-mp : pour le support de la bibliothèque **Lua** **mplib** ;

minim-math : pour la gestion des mathématiques **Unicode** ;

minim-pdf : pour la gestion des liens hypertextes et les ancrages **PDF** ;

minim-xmp : pour la gestion des métadonnées **PDF**.

Les noms de ces packages sont explicites : il s'agit d'être minimal tout en offrant beaucoup d'outils.

C'est au premier d'entre eux, **minim-mp**, que nous nous intéressons ici. Il permet donc l'intégration de la bibliothèque **Lua** **mplib** pour le format plain **T_EX**. Ce package **T_EX** fournit la commande de base **\directmetapost** qui permet d'exécuter du code **MetaPost** directement dans le code **L^AT_EX** (un peu comme **\directlua** permet d'exécuter du code **Lua**). Chaque invocation de la commande **\directmetapost** lance une instance **MetaPost** indépendante. Ainsi, nous obtenons un dessin **MetaPost** avec le code suivant :

78. Le calcul scientifique est une discipline aux contours flous, qui regroupe un ensemble de champs mathématiques et informatiques permettant la simulation numérique des phénomènes modélisés par des équations mathématiques.

Exemple 18*code*

```
1 Traçons un cercle : \directmetapost{
2 beginfig(1);draw fullcircle scaled 1cm;endfig;}.
```

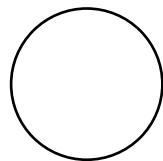
résultat

Traçons un cercle : .

Cette commande produit, pour chaque groupe MetaPost délimité par `beginfig` et `endfig`, des boîtes que nous pouvons manipuler.

Exemple 19*code*

```
\scalebox{2}{\directmetapost{beginfig(2);draw
fullcircle scaled 1cm;endfig;}}
```

résultat

Mais `minim-mp` n'est pas qu'un package `TEX` : c'est aussi un package `LATEX` qui, bien qu'expérimental pour l'instant, est tout à fait utilisable⁷⁹. Celui-ci nous fournit l'environnement `metapost` pour exécuter du code MetaPost.

Exemple 20

```
\begin{metapost}
beginfig(1);
draw (0,0)--(1cm,1cm);
endfig;
\end{metapost}
```

Le package `LATEX` peut même être chargé avec l'option `luamplib` et fournit alors des commandes et environnements analogues à ceux du package `luamplib`.

L'environnement `metapost` crée des instances MetaPost indépendantes, mais l'utilisateur peut aisément définir de nouveaux environnements qui, eux, sont persistants. Dans le code suivant, nous créons un environnement `mppersist` qui permet de garder en mémoire les exécutions précédentes des environnements `mppersist`.

⁷⁹. La rédaction de cet article a d'ailleurs permis de faire remonter quelques dysfonctionnements, très vite corrigés par l'auteur : <https://gitlab.com/renkema/minim/-/issues>.

Exemple 21*code*

```

1 \newmetaenvironment{mpersist}
2   \begin{mpersist}
3     a:=1cm;
4   \end{mpersist}
5
6   \begin{mpersist}
7     beginfig(1);
8     draw (0,0)--(a,a);
9     endfig;
10    \end{mpersist}
```

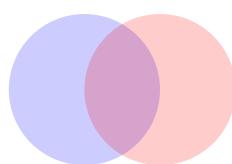
résultat

Bien que la documentation de `minim-mp` manque de détails et d'exemples, ce package présente quelques avantages par rapport à `luamplib`. Par exemple, la commande `\mpcolor`, qui permet d'utiliser une couleur `LATEX` dans le code MetaPost, ne peut, avec `luamplib`, être utilisée qu'avec l'opérateur `withcolor`. Tandis que `minim-mp` permet de l'utiliser exactement comme une couleur MetaPost normale. De plus, `minim-mp` permet d'utiliser des fonctionnalités du format `PDF` directement depuis MetaPost. Par exemple, alors que MetaPost, utilisé comme programme indépendant de `LuATEX`, n'a nativement pas de fonction de transparence⁸⁰, `minim-mp` fournit plusieurs commandes de gestion de la transparence dont l'opérateur `withalpha()`.

Exemple 22*code*

```

1 \begin{metapost}
2 beginfig(4);
3 fill fullcircle scaled 2cm withcolor blue withalpha (0.
4   2);
5 fill fullcircle scaled 2cm shifted (1cm,0) withcolor
6   red withalpha(0.2);
7 endfig;
8 \end{metapost}
```

résultat

Enfin, `minim-mp` donne accès à de nombreux paramétrages et personnalisations pour qui accepte de se plonger dans ses arcanes.

⁸⁰. Il faut recourir à des constructions comme l'a proposé Anthony Phan : <http://www-math.univ-poitiers.fr/~phan/metalpha.html>.

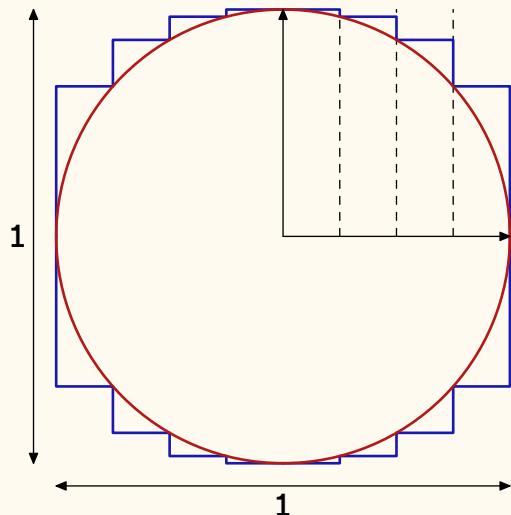
Avec le package **animate**

Les méthodes numériques de calcul reposent bien souvent sur le calcul approché d'un objet mathématique, soit par discrétisation d'un élément continu, soit par troncature d'une somme infinie. Ainsi, on aura souvent un paramètre fixé, que l'on peut noter N , tel que quand N tend vers l'infini, alors notre méthode numérique (dépendante de ce paramètre N) doit converger vers l'objet mathématique que l'on souhaite calculer. Évidemment, ces concepts ont des définitions mathématiques rigoureuses que nous n'aborderons pas ici.

Une première approche... et $\pi = 4$!

La première méthode numérique que nous avons souhaité présenter aux élèves est une mauvaise méthode⁸¹ qui semble fonctionner et qui semble prouver que $\pi = 4$. On va considérer un quart de cercle, et par symétries, on réalisera sur tout le cercle les opérations décrites. Ainsi, pour N fixé, on divise le rayon (suivant l'axe des abscisses) en N parties égales, puis on projette verticalement (vers le haut) les points obtenus sur le cercle. Une fois ces points obtenus, on les relie, extérieurement au cercle, par des lignes brisées constituées de segments uniquement horizontaux ou verticaux. Le dessin de la figure 20 éclaire cette explication.

FIGURE 20 – Illustration de la mauvaise méthode numérique qui semble montrer que $\pi = 4$. Ici $N = 4$.



Utiliser un langage de programmation pour produire ce type d'image permet de paramétriser le dessin, et donc d'illustrer aisément l'évolution d'une méthode numérique.

MetaPost nous permet de programmer plus aisément que **TikZ** car il est plus proche d'un langage de programmation classique. Voici un code MetaPost, que nous ne détaillerons pas, produisant l'image décrite précédemment :

⁸¹. L'intérêt est double : illustrer sur un exemple simple le fait d'avoir une méthode qui semble se rapprocher de ce qu'on souhaite calculer quand le paramètre N tend vers l'infini et illustrer que les choses qui semblent évidentes en mathématiques doivent être considérées rigoureusement.

Exemple 23

```

1 N:=4; % le paramètre
2 color DarkBlue,DarkRed,DarkGreen; % trois couleurs
3 DarkBlue := (0.1,0.1,0.7);
4 DarkRed := (0.7,0.1,0.1);
5 DarkGreen := (0.1,0.7,0.1);
6 beginfig(1);
7 u:=3cm; % l'unité
8 pair Point, pointS;
9 path poly,droite,cercle;
10 cercle := fullcircle scaled 2u; % on définit le cercle
% on légende les dimensions
11 drawdblarrow (-1.1u,-u)--(-1.1u,u);
12 drawdblarrow (-u,-1.1u)--(u,-1.1u);
13 label.lft(btex 1 etex, (-1.1u,0));
14 label.bot(btex 1 etex, (0,-1.1u));
15 if(N=1): % si N=1 alors le polygone est le carré
16   draw unitsquare scaled 2u shifted (-u,-u)
17   withcolor DarkBlue withpen pencircle scaled 1pt;
18 else: % sinon
19   poly:=(0,u);
20   pointS := (0,u);
21   for j:=1 upto (N-1): % pour tous les points du
22     rayon subdivisé
23     % on définit le segment vertical passant par
24     % le point
25     xp := j*u/(N);
26     droite := (xp,0)--(xp,u);
27     if(N<=10): % on ne trace les pointillés de ces
28       segments que pour N<=10
29       draw droite dashed evenly;
30     fi
31     % on définit le point sur le cercle comme
32     % l'intersection entre le cercle et le segment
33     Point := droite intersectionpoint cercle;
34     % rajout du segment vertical au polygone
35     poly := poly -- (xp,ypart pointS)--Point;
36     pointS := Point; % sauvegarde du point courant
37   endfor
38   % dernier point
39   poly := poly -- (u,ypart pointS)--(u,0);
40   % tracé du polygone
41   draw poly withcolor DarkBlue withpen pencircle
42   scaled
43   1pt;
44   % et des trois réflexions
45   draw poly reflectedabout((0,0),(u,0)) withcolor
46   DarkBlue withpen pencircle scaled
47   1pt;

```

```

44      draw poly reflectedabout((0,0),(0,u)) withcolor
45      DarkBlue withpen pencircle scaled
46          1pt;
47      draw poly rotated 180 withcolor DarkBlue withpen
48      pencircle scaled
49          1pt;
50  fi
51 % on dessine le cercle
52 draw cercle withcolor DarkRed withpen pencircle scaled
53          1pt;
54 % on trace les axes
55 drawarrow (0,0)--(u,0);
56 drawarrow (0,0)--(0,u);
57 endfig;

```

Il est possible d'encapsuler ce code dans une macro \LaTeX pour l'appeler simplement avec le package `animate` et produire ainsi une animation⁸².

Exemple 24

```

1 \newcommand{\piquatrel}[1]{%
2 \begin{metapost}
3 N:=#1; % le paramètre
4 color DarkBlue,DarkRed,DarkGreen; % trois couleurs
5 DarkBlue := (0.1,0.1,0.7);
6 DarkRed := (0.7,0.1,0.1);
7 % ... code MetaPost précédent
8 endfig;
9 \end{metapost}
10 }

```

Une fois cette macro définie, il suffit de l'insérer dans l'environnement `animateinline` :

Exemple 25

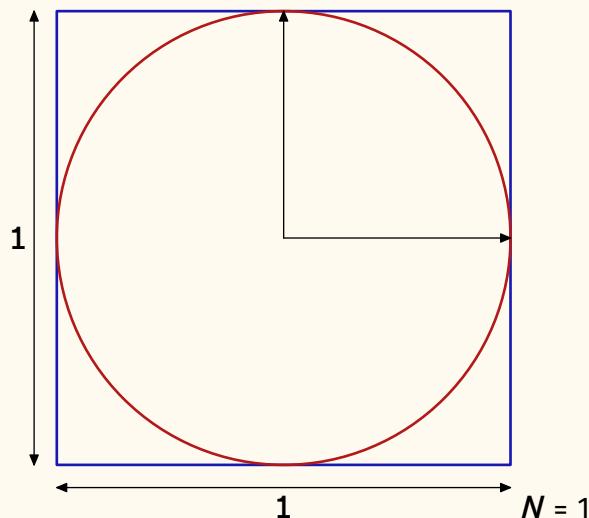
```

1 \begin{animateinline}[\controls]{1}%
2   \multiframe{20}{N=1+1}{%
3     \piquatrel{\N}%
4     $N=\N$%
5   }%
6 \end{animateinline}%

```

... pour obtenir le résultat suivant :

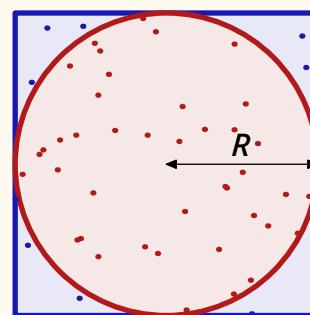
82. Visionnable par exemple avec Okular ou Acrobat Reader.



La production de ces images est un vrai plaisir mais cette méthode numérique semble montrer que $\pi = 4$! Pourriez-vous expliquer pourquoi cette méthode ne montre pas ce qu'elle semble montrer ?

Un peu d'aléatoire

La dernière méthode numérique⁸³ que nous avons présentée aux élèves de troisième est celle de Monte Carlo, simple à comprendre : on considère un cercle et son carré circonscrit, et on tire N points aléatoirement dans le carré.



On compte alors les points qui se trouvent à l'intérieur du cercle et on note ce nombre n . On calcule ensuite :

$$\frac{n}{N} \times 4,$$

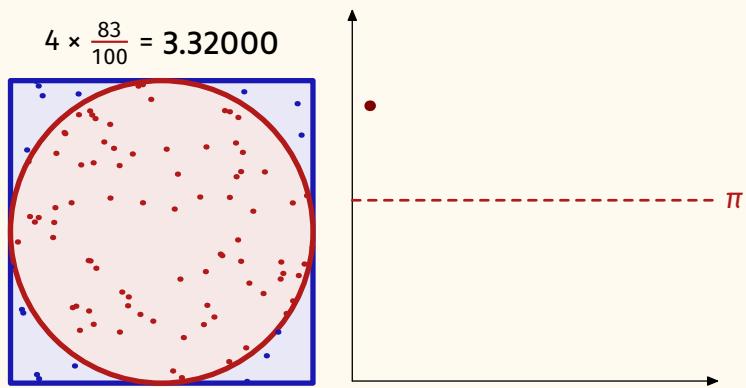
qui donne une approximation de π .

En effet, en mathématiques, un résultat remarquable, la *loi des grands nombres*, garantit que :

$$\text{Probabilité d'être dans le cercle} = \frac{\text{Aire cercle}}{\text{Aire carré}} = \frac{\pi R^2}{(2R)^2} = \frac{\pi}{4}$$

Grâce à LualATEX, MetaPost et [animate](#), nous pouvons illustrer ce phénomène :

⁸³. Avant cela, nous avons présenté la méthode d'Archimède de construction de polygones réguliers inscrits et circonscrits au cercle, mais les codes LATEX et MetaPost restent très proches de ce qui a déjà été présenté ici.



Pour ce faire, nous définissons un environnement persistant pour que le code MetaPost exécuté dans cet environnement soit visible par les autres environnements.

Exemple 26

```
1 \newmetapostenvironment[mathmode='double']{
    mpmontecarlo}
```

Il est nécessaire d'ajouter `mathmode='double'` en option de la commande pour utiliser la représentation numérique en double précision et ainsi pouvoir utiliser des nombres entiers très grands (ici, on va en générer 6000).

Ensuite, nous générerons de nombreux points dans le carré et définissons la fonction MetaPost qui calcule le rapport entre le nombre de points à l'intérieur du cercle et le nombre de points total. Cela se fait avec le code suivant à inclure dans un environnement `mpmontecarlo` :

Exemple 27

```
1 numeric monteC[];
2 pair Points[];
3
4 for i:=1 upto 6000:
5     Points[i] := (uniformdeviate(4cm)-2cm,
6                     uniformdeviate(4cm)-2cm);
7 endfor
8
9 vardef proba(expr N)=
10    save MC;
11    MC := 0;
12    for i:=1 upto N:
13        if(abs(Points[i])<2cm):
14            MC := MC + 1/N;
15        fi
16    endfor
17    (4*MC)
18 enddef;
```

La figure représentant le nuage de points dans le carré circonscrit au cercle

est obtenue à l'aide du code MetaPost suivant :

Exemple 28

```

1 color DarkBlue,DarkRed,DarkGreen; % trois couleurs
2 DarkBlue := (0.1,0.1,0.7);
3 DarkRed := (0.7,0.1,0.1);
4 DarkGreen := (0.1,0.7,0.1);
5 N:=#1; % le paramètre qui proviendra de l'argument d'
       une commande LaTeX
6 beginfig(1+#1);
7   % le carré
8   fill unitsquare scaled 4cm shifted (-2cm,-2cm)
9   withcolor 0.9[DarkBlue,white] ;
10  draw unitsquare scaled 4cm shifted (-2cm,-2cm)
11   withcolor DarkBlue withpen pencircle scaled 2pt;
12   % le cercle
13   fill fullcircle scaled 4cm withcolor 0.9[DarkRed,
14   white];
15   draw fullcircle scaled 4cm withcolor DarkRed
16   withpen pencircle scaled 2pt;
17   % on va compter les points à l'intérieur
18   % et les tracer suivant s'ils sont dedans ou non
19   Ni:=0;
20   for i:=1 upto (N):
21     if(abs(Points[i])<2cm):
22       drawdot Points[i] withpen pencircle scaled
23       2pt withcolor
24       DarkRed;
25       Ni:=Ni+1;
26     else:
27       drawdot Points[i] withpen pencircle scaled
28       2pt withcolor DarkBlue;
29     fi
30   endfor
31   % le label
32   string labi,v;
33   labi := "$4\times \frac{" &decimal Ni &}{" &decimal
34   N &}=$";
35   v:=substring (0,7) of decimal proba(N);
36   label.lft(texttext(labi),(0,2.5cm));
37   label.rt(texttext(v),(-0.1cm,2.5cm));
38 endfig;
```

Pour utiliser ce code avec `animate`, nous l'encapsulons dans une commande `LATEX` :

Exemple 29

```

1 \newcommand\montecarlo[1]{%
2   \begin{mpmontecarlo}
```

```

3      color DarkBlue,DarkRed,DarkGreen; % trois couleurs
4      DarkBlue := (0.1,0.1,0.7);
5      DarkRed := (0.7,0.1,0.1);
6      DarkGreen := (0.1,0.7,0.1);
7      N:=#1;
8      % code MetaPost précédent
9      %...
10     \end{mpmontecarlo}
11 }
```

Avec le code MetaPost suivant, nous procédons de la même manière pour tracer la courbe d'évolution du calcul suivant la valeur de N :

Exemple 30

```

1 beginfig(#1);
2   % unités
3   uX := 0.23cm;
4   uY := 7cm;
5   % bornes
6   ymin:=2.8;
7   ymax:=3.5;
8   xmax:=20;
9   % axes
10  drawarrow (-1uX,ymin*uY)--(xmax*uX,ymin*uY);
11  drawarrow (-1uX,ymin*uY)--(-1uX,ymax*uY);
12  % la droite  $y=\pi$ 
13  draw ((-1uX,3.1415uY)--(xmax*uX,3.1415uY)) dashed evenly
14  withcolor DarkRed withpen pencircle scaled 1pt;
15  label.rt(btex $\pi$ etex,(xmax*uX,3.1415uY))
16  withcolor DarkRed;
17  % on fabrique la ligne brisée du graphique
18  path plot;
19  i:=0;
20  for j:=100 step 300 until #1:
21    if(j=100):
22      plot:=(i*uX,proba(j)*uY);
23    else:
24      plot:=plot--(i*uX,proba(j)*uY);
25    fi
26    drawdot (i*uX,proba(j)*uY) withpen
27    pencircle
28      scaled 4pt withcolor 0.5[red,black];
29    i:=i+1;
30  endfor
31  % on trace la ligne
32  draw plot withcolor 0.5[red,black];
33 endfig;
```

Et comme précédemment, nous l'encapsulons dans une commande L^AT_EX :

Exemple 31

```

1 \newcommand{\suite}[1]{%
2   \begin{mpmontecarlo}
3     beginfig(#1);
4     uX := 0.23cm;
5     uY := 7cm;
6     % code MetaPost précédent
7     % ...
8     endfig;
9   \end{mpmontecarlo}
10 }
```

Avec ces commandes \LaTeX , nous pouvons donc utiliser le package `animate` pour produire l'animation présentée ci-dessus.

Exemple 32

```

1 \begin{center}
2   \begin{animateinline}[controls]{1}%
3     \multiframe{15}{N=100+300}%
4       \begin{tabular}{cc}
5         \montecarlo{\N} & \suite{\N} \\
6         \end{tabular}
7     }%
8   \end{animateinline}%
9 \end{center}
```

Conclusion

\LaTeX et MetaPost me permettent de produire des choses comme je les imagine et cela très simplement. Pour la plupart de mes productions, les aspects de programmation et de géométrie de ces logiciels me rendent de grands services. Dans les deux exemples que j'ai montrés ici, ce fut en direction de collégiens et de collégiennes, et s'ils n'ont pas fait de retour explicite sur l'aspect des diapositives, ils ont, semble-t-il, apprécié l'exposé : je pense que les supports n'y sont pas pour rien. Au passage, alors que j'avais utilisé `luamplib` pour produire mes diapositives `beamer`, j'ai profité de l'écriture de cet article pour essayer `minim-mp` qui est une excellente alternative permettant, grâce à `Lua\TeX`, d'utiliser du code MetaPost directement dans un source $(\text{\La})\TeX$. Vive \LaTeX , vive MetaPost !

Maxime Chupin



♪ \dante_tutorial:nn{expl3}{2022}

Nous publions ici, sous son titre original, un article de Marei Peischl originellement paru fin 2022 dans Die TeXnische Komödie⁸⁴. Il nous a semblé avoir un grand intérêt. Notons que cet article est également paru en anglais, traduit par son autrice, dans le premier numéro du TUGboat de l'année 2023⁸⁵.

De ce que nous en savons, le premier contact avec `expl3` – la syntaxe de la couche de programmation de `LATEX3` – est souvent effrayante et trompeuse :

```

1  \ExplSyntaxOn
2  \clist_map_inline:nn \l_tmpa_clist {
3      \__ptxcd_add_item:n {#1}
4  }
5  \ExplSyntaxOff

```

`LATEX3` n'est plus une nouveauté logicielle que la communauté attendait depuis des décennies. `LATEX3` est parmi nous depuis belle lurette, il est utilisé par de nombreuses personnes sans pour autant être vraiment remarqué.

La couche de programmation de `LATEX3` constitue le socle de cet ensemble. Elle offre une interface normalisée que les développeurs de paquets comme les utilisateurs peuvent utiliser, directement ou non, que ce soit pour programmer des mécanismes complexes ou pour gérer du contenu de manière nettement plus flexible qu'avec des fichiers `LATEX` classiques.

En définitive, les objectifs les plus importants de `LATEX3` sont :

- une interface unifiée pour les fonctions et variables ;
- une syntaxe modernisée ;
- une simplification du contrôle de l'expansion...

... ce qui permet de programmer très facilement avec `LATEX`.

Programmer se révèle utile lorsque, en appliquant certains paramètres, on modifie la mise en page ou la structure générale d'un document. Un exemple classique nous vient de l'enseignement, où l'on compose un sujet d'examen en dissimulant ou non les solutions des exercices. Un autre exemple est l'utilisation de données externes – typiquement une liste⁸⁶, mise en forme de manière automatisée :

Exemple 34

code

```

1  \ExplSyntaxOn
2  \begin{enumerate}

```

84. http://archiv.dante.de/DTK/PDF/komoedie_2022_4.pdf

85. <https://doi.org/10.47397/tb/44-1/tb136peischl-expl3>

86. Note du traducteur – c'est volontairement que, dans la liste en question, nous n'avons pas traduit *eins*, *zwei*, *drei* : ils seront utilisés plus loin pour illustrer les capacités d'expansion de `expl3`. Or, de l'expansion à la récursivité, il n'y a qu'un pas, et nous utilisons l'éditeur de texte `emacs`, qui utilise le langage `Lisp`... nous nous souvenons des deux machines `Lisp` du MIT, le *Massachusetts Institute of Technology*. La première était appelée EINE, sigle récursif pour *Eine Is Not Emacs*, et la seconde ZWEI, pour *Zwei Was Eine Initially*. Nous manquons hélas d'informations sur DREI.

	code (suite)
3	\clist_map_inline:nn { eins, zwei, drei } { \item #1 }
4	\end{enumerate}
5	\ExplSyntaxOff
	résultat
	<ol style="list-style-type: none"> 1. eins 2. zwei 3. drei

La syntaxe `expl3` paraît cryptique à des yeux habitués à d'autres langages de programmation. Les deux-points mêlés à des tirets bas et les conventions de nommage en vigueur créent une structure syntaxique singulière, qui trouve son origine dans le fait que L^AT_EX, et par conséquent `expl3`, n'est rien d'autre qu'un langage de macro-commandes. Les chaînes de caractères y sont remplacées par ce qu'elles désignent et non par des opérations réelles, comme c'est le cas dans les langages de scripts.

Pour comprendre la structure d'`expl3`, il est donc nécessaire de comprendre les bases de la macro-expansion et du concept de *catcode* (pour *category code*). C'est pourquoi les sections suivantes, qui décrivent la structure syntaxique de ce langage, explicitent ces concepts. Si ces derniers sont déjà familiers au lecteur, il lui suffit de sauter les sections correspondantes.

Commutation syntaxique avec T_EX, L^AT_EX et `expl3`

Quand T_EX compile des données, il ne fait pas que lire une suite de caractères : il attribue une catégorie à chacun d'entre eux. Chaque catégorie détermine l'usage qui sera fait du caractère en question. L'attribution d'une catégorie à un caractère se fait via les fameux *catcodes* (abréviation de *category codes*, en anglais). À chaque caractère correspond un code de caractère, et à celui-ci est ensuite temporairement attribué un code de catégorie.

Au total, T_EX définit seize catégories différentes. Toutefois, l'affectation d'une catégorie à un caractère peut varier à l'intérieur d'un même document. Le cas le plus courant est une variation en fonction d'une langue donnée, par exemple en utilisant `babel` [1]. C'est ainsi que la chaîne « "a » devient « ä » en allemand tandis que ces deux caractères seront traités séparément en français. Le comportement en allemand est dû à la catégorie 13, « active » ; les caractères actifs ne sont plus de simples caractères, mais des instructions – dans le cas du « "a », placer un tréma sur la lettre qui suit l'instruction. En français, aucun des deux caractères n'est placé dans la catégorie 13 : chacun d'entre eux est lu en tant que tel, d'où le résultat : « "a ».

La liste suivante présente toutes les catégories disponibles, avec des explications et des exemples qui vous seront probablement familiers.

0. Caractère d'échappement (\)
1. Début de groupe ({)
2. Fin de groupe (})
3. Passage en mode mathématique (\$)

4. Alignement ($\&$)
5. Fin de ligne ($\langle return \rangle$)
6. Paramètre de commande (#)
7. Exposant mathématique (^)
8. Indice mathématique : (_)
9. Caractère ignoré ($\langle null \rangle$)
10. Espace ($_$)
11. Lettre (l'alphabet : A, B... a, b... – et, seulement avec X_ET_EX et L_AT_EX, leurs versions diacritées)
12. « Autre » caractère (tout le reste : ., 1, :, @, etc.)
13. Caractère actif, à interpréter comme une séquence de contrôle (~)
14. Début de commentaire (%)
15. Caractère invalide : ($\langle delete \rangle$)

Le caractère @ dans les macro-commandes de T_EX et L_AT_EX

T_EX utilise le caractère @ pour protéger ses macros privées : cela empêche les utilisateurs finaux d'y accéder. Ce n'est pas sans raison qu'elles sont ainsi protégées. Afin d'éviter des effets secondaires inattendus, d'éventuelles modifications doivent donc être effectuées avec prudence. Il est nécessaire de modifier la catégorie du caractère @ vers la classe 11 afin de pouvoir l'utiliser dans les noms de macros.

Pour ce faire, on utilise des commutateurs :

```
1 \makeatletter
2 \makeatother
```

Un exemple classique de l'utilisation du caractère @ est la création de variantes étoilées de certaines commandes. Pour cela, il faut définir en interne deux macros auxiliaires, qui elles-mêmes sont souvent protégées par le signe @.

```
1 \makeatletter
2 \newcommand*{\cmd}{\@ifstar\@cmdstar\@cmd}
3 \newcommand*{\@cmd}{sans *}
4 \newcommand*{\@cmdstar}{avec *}
5 \makeatother
```

La macro donne alors le résultat suivant :

Exemple 37

	code	résultat
1	\cmd\\cmd*	sans * avec *

La syntaxe expl3

Comme la syntaxe de expl3 est conçue pour la programmation, L_AT_EX se comporte ici de manière fondamentalement différente de ce qui serait judicieux pour une sortie de texte.

- Les espaces et les retours à la ligne dans le code délimitent les chaînes de caractères ; autrement, ils sont ignorés.
- Les lignes vides ne marquent pas une fin de paragraphe.
- Les deux points (:) et le tiret bas (_) font partie intégrante des noms de macro.
- Il y a une distinction entre les fonctions et les variables.
- Il est recommandé de mettre des espaces autour des accolades si elles ne contiennent pas qu'un seul paramètre.

Ces modifications permettent de structurer un peu mieux le code, sans en modifier le sens. La syntaxe est introduite par un commutateur, analogue dans son utilisation au bien connu `\makeatletter` :

```
1 \ExplSyntaxOn
2 \ExplSyntaxOff
```

(commutateur syntaxique)

Une autre caractéristique typique de `LATEX3` est l'utilisation de majuscules et de minuscules dans ses noms de macro. La lisibilité du code s'en trouve améliorée, y compris au sein de la syntaxe `LATEX` classique.

La structure des noms

La structure des noms suffit à `expl3` pour distinguer entre les fonctions, qui traitent des contenus/arguments, et les variables, qui ne font que stocker une valeur.

```
1 \Module_Description:Arguments
```

(fonctions)

```
2 \Validité_Module_Description_Type-de-donnée
```

(variables)

Les variables

Les variables stockent des valeurs. `expl3` fournit pour cela différents types de données. En revanche, la structure du nom de chacun de ces types de données est identique. Techniquelement, il s'agit d'une convention. Toutefois, le respect de ces règles permet aux utilisateurs de comprendre plus facilement le code.

```
1 \Validité_Module_Description_Type-de-donnée
```

(variables)

Validité : constante (c), globale (g) ou locale (l).

Privée : les variables privées, qui ne doivent pas être utilisées par les utilisateurs finaux, présentent deux tirets bas entre l'indication de validité et le nom du module. Les variables normales n'en ont qu'un.

Module : nom du package⁸⁷, ceci pour éviter les doublons. L'enregistrement de ces noms se fait selon le processus décrit dans [2].

Description : que stocke la variable et pourquoi ?

87. NdT : ou de la classe.

Type de donnée : comment les différentes valeurs sont-elles enregistrées dans la variable ? Comment doit-elle être traitée ?

Un exemple de variable privée est la liste de *tokens*, c'est-à-dire une liste de caractères.

1 \l_siunitx_complex_sign_tl

Tous les types de données sont associés à un module. Les listes de *tokens*⁸⁸ proviennent du module `l3tl`⁸⁹. D'autres types de données ont une autre désignation correspondant à cette abréviation.

Toutes les interfaces sont documentées en [3]. La liste suivante en donne un petit échantillon :

tl	<i>tokenlist</i> , chaîne de caractères qui ont encore différents <i>catcodes</i>
str	<i>string</i> , chaîne de caractères comprenant seulement des lettres, des espaces ou des caractères de catégorie 12 (autres)
bool	<i>boolean</i> , valeurs de vérité
box	boîte
coffin	« boîte avec poignées », une boîte dotée d'ancrages permettant un placement relatif à d'autres objets
dim	dimension
skip	dimension extensible
fp	<i>floating point</i> , nombres à virgule flottante
int	<i>integer</i> , nombre entier
prop	<i>property list</i> , liste de clés/valeurs
seq	<i>sequence</i> , séquence, suite
clist	liste de valeurs séparées par des virgules
stream	flux d'entrée/sortie pour la lecture/écriture de fichiers

Les fonctions

Le terme de fonction est peut-être ici trompeur par rapport aux langages de programmation. `TeX`, et donc `expl3`, est purement un langage de traitement

88. NdT : ... ou lexème en français.

89. Note de Denis Bitouzé⁹⁰ : les *modules* dont il est question dans ce paragraphe sont ceux de `expl3` et en constituent en quelque sorte des bibliothèques, chacune d'elles dédiée à un type de donnée. Ainsi, `l3tl` est-il celui dédié aux *tokenlist*, `l3int` est-il celui dédié aux *integers* (entiers). Ces *modules* sont à ne pas confondre avec ceux dont il est question ci-dessus et qui sont le nom du package ou de la classe que l'auteur est en train de créer ; ainsi le code du package `siunitx`, programmé en `LATeX3`, contient-il de nombreuses variables de la forme `\Validité_siunitx_Description_Type-de-donnée`, par exemple `\l_siunitx_complex_sign_tl`.

90. Le lectorat appréciera l'intervention de Denis Bitouzé, qui témoigne du sérieux de notre entreprise : la *Lettre GUTenberg* est certes irrégulomadaire et rédigée par des bénévoles, mais ce sont des bénévoles consciencieux, qui se relisent les uns les autres et n'hésitent pas à amender un texte, en l'espèce dans un but pédagogique. Voir la liste des impétrants page 113. Au sujet de l'intrication des notes, voir... la note 112 en page 100 !

de macros qui donne l'impression d'avoir des fonctions alors qu'il ne fait que substituer des chaînes de caractères. Néanmoins, dans `expl3`, on peut aussi considérer que les fonctions traitent leur contenu, si ce n'est qu'il n'y a pas de valeur de sortie au sens des langages de programmation traditionnels : la « valeur de sortie » des fonctions est envoyée dans le flux d'entrée – et donc écrite dans le document.

La convention de nommage des fonctions est la suivante :

```
1 \Module_Description:Spécifications-des-arguments
```

Le **Module** et la **Description** nom sont identiques aux mêmes champs que nous avons déjà décrits pour les variables. La différence entre une fonction locale et une fonction globale est toutefois inscrite dans la description. Par convention, on insère le mot `set` devant une affectation (locale) et on y ajoute un `g` pour une affectation globale : `gset`. Les fonctions permettant de définir des variables entières en sont des exemples :

```
1 \int_set:Nn
2 \int_gset:Nn
```

Avec la syntaxe `expl3`, les arguments attendus sont indiqués après les deux points. Ainsi, les deux exemples ci-dessus attendent chacun deux arguments : un de type `N` et un de type `n`. On peut ainsi voir directement le nombre et le type des arguments attendus par une fonction. Cela sera important plus tard pour le contrôle du développement des macros, mais pour le moment, nous nous limitons aux types `N/n`. La lettre `n` (majuscule ou minuscule) signifie *No manipulation*⁹². L'argument est donc transmis à la fonction sans autre traitement.

L'usage de la majuscule ou de la minuscule indique si la fonction attend ici un *token* unique ou un argument groupé. Les majuscules correspondent à des *tokens*. Les minuscules correspondent à des groupes. La spécification d'argument `:Nn` déclare que la fonction attend d'abord un *token* et ensuite un groupe. Ainsi, la macro `\int_set:Nn` pourrait donc être utilisée comme suit :

```
1 \int_set:Nn \l_tmpa_int { 5 }
```

L'exemple ci-dessus donne à une variable entière (premier argument, dans ce cas `\l_tmpa_int`) la valeur 5 (donnée dans un groupe).

La fonction est donc très similaire à la classique `\setcounter` `LATEX`, mais l'argument de `\int_set:Nn` peut également être calculé. La fonction `\int_use:N` donne la valeur de la variable donnée en argument, et dans l'exemple qui suit, l'imprime.

Cf. exemple 45 page suivante.

92. NdT : Pas de manipulation.

Exemple 45 (cf. page 74)

```

1 \ExplSyntaxOn
2 \int_set:Nn \l_tmpa_int { 5 + 2 * 3 }
3 \int_use:N \l_tmpa_int
4 \ExplSyntaxOff

```

code

11

résultat

Un exemple de type de données : l3dim

Cette section est consacrée au type de données identifiées par dim dans la liste précédente, et servant pour les longueurs. Les fonctions les plus courantes y sont abordées. Il existe des fonctions similaires pour d'autres types de données⁸⁹. On trouvera un passage en revue complet dans [3].

Création et initialisation – type d'argument N/n

```

1 \dim_new:N
2 \dim_const:Nn

```

L'instruction `\dim_new:N` crée une nouvelle variable. Celle-ci existe alors globalement, mais elle peut aussi n'exister que localement. La convention de nommage est ici importante. Si nous souhaitons créer une variable locale et une variable globale pour le tutoriel, nous pouvons le faire ainsi :

```

1 \dim_new:N \l_dante_test_dim
2 \dim_new:N \g_dante_test_dim

```

Il est donc possible qu'il y ait deux variables avec la même description, dont l'une est définie localement et l'autre globalement.

Quand on crée une constante, on renseigne directement sa valeur lors de la déclaration :

```

1 \dim_const:Nn \c_dante_test_dim { 5cm }

```

Pour les variables, qui sont donc modifiables, l'affectation se fait séparément :

```

1 \dim_set:Nn \l_dante_test_dim { 3cm }
2 \dim_gset:Nn \g_dante_test_dim { 1cm + 5mm }

```

Comme pour les nombres, il est également possible de réaliser des calculs au moment de l'affectation. Il est important de noter que la précision est limitée à celle de TeX pour les longueurs. La plus petite unité est ici 1sp = 0,000 02 pt. Ce qui est définitivement assez petit pour avoir une précision plus que suffisante pour les productions imprimées.

Outre l'affectation, il existe également des commandes pour l'addition et la soustraction de longueurs. La structure des noms reste identique. Tous les détails peuvent être consultés dans [3].

Requêtes et boucles – type d'argument T/T/TF

Lors du traitement des variables, il est également possible d'en comparer les valeurs et, le cas échéant, de créer des boucles à partir de celles-ci. Pour les longueurs, la commande la plus simple est la suivante :

```
1 \dim_compare:nTF
```

Cette macro est donc utilisée pour comparer des longueurs. Les opérateurs de comparaison sont :

égalité	= ou ==
supériorité ou égalité	>=
supériorité	>
infériorité ou égalité	<=
infériorité	<
inégalité	!=

Il existe quelques variantes de la fonction `\dim_compare` qui n'autorisent qu'une partie de ces opérateurs, ce qui permet de réduire les temps de traitement. Nous présentons ici la variante qui s'utilise le plus facilement. Comparons les longueurs que nous venons de créer :

Exemple 51

code

```
1 \ExplSyntaxOn
2 \dim_compare:nTF
3 { \g_dante_test_dim >= \l_dante_test_dim }
4 { est~supérieur~ou~égal }
5 { est~inférieur }
6 \ExplSyntaxOff
```

résultat

est inférieur

Comme décrit plus haut, chacune des lettres du type d'argument TF correspond à un argument, qui peut être dans ce cas aussi bien un groupe qu'un *token*⁹³. T et F signifient respectivement *true* et *false*. Mais le langage `expl3` présente ici une particularité : si une seule réponse est attendue, il suffit d'omettre les types d'argument inutilisés – ici le T !

Exemple 52

code

```
1 \ExplSyntaxOn
2 \dim_compare:nF
3 { \g_dante_test_dim > \l_dante_test_dim }
4 { est~non~supérieur }
5 \ExplSyntaxOff
```

93. NdT :: Contrairement au couple N/n, T et F sont toujours écrits en majuscule.

est non supérieur

résultat (suite)

Sorties pour le débogage

Lors d'un travail complexe de programmation, il est utile de vérifier de temps à autre que le traitement des valeurs se déroule correctement. Pour ce faire, `expl3` offre des commandes qui permettent d'afficher dans le terminal l'état actuel d'une variable ou de l'écrire dans un fichier journal.

```
1 \dim_show:N
2 \dim_show:n
3 \dim_log:N
4 \dim_log:n
```

Les macros présentant le type d'argument `n` donnent une expression de longueur et non une variable de longueur. Cela permet donc d'y effectuer des calculs ou d'y utiliser une macro, contenant par exemple une indication en centimètres, qui sera évaluée.

Spécification des types d'arguments de fonction

Outre les types d'arguments exposés plus haut, il en existe quelques autres, dont le comportement diffère de celui des arguments `LATEX` classiques.

Type	Nom et description
<code>N/n</code>	<i>no manipulation</i>
<code>TF/T/F</code>	<i>true/false</i>
<code>c</code>	<i>csname</i>
<code>V/v</code>	<i>value</i>
<code>o</code>	<i>expand once</i>
<code>e</code>	<i>exhaustive expansion</i> (et si besoin la macro elle-même est développable)
<code>x</code>	<i>e xhaustive expansion</i>
<code>f</code>	<i>full expansion</i>
<code>p</code>	<i>paramètre</i>
<code>w</code>	<i>weird</i>

`N` et `TF` ont été exposés ci-dessus. Nous en venons au type d'argument `c`. Les types `o`, `f`, `x`, `e` et `V` viendront ensuite. Les types `p` et `w` sont rarement utilisés par les néophytes ; nous n'en parlerons donc pas. Ils n'ont été mentionnés que par souci d'exhaustivité. On trouvera plus d'informations à leur sujet dans [4], par exemple.

`csname/endcsname` – type d'argument `c`

L'un des principes fondamentaux de `LATEX` est la possibilité de créer des noms de macro de manière dynamique.

```
1 \csname nom\endcsname
```

Si l'on s'en tient à l'aspect visuel, l'environnement `\csname/\endcsname` ne fait rien d'autre que d'insérer une contre-oblique (`\`) avant le contenu dudit environnement :

Exemple 55

	code	résultat
1	<code>\LaTeX{}</code>	<code>\LaTeX = \LaTeX</code>
2	=	
3	<code>\csname LaTeX\endcsname</code>	

Les références internes en sont un exemple classique d'utilisation. Ici, la macro `\label` crée une macro interne en lui donnant comme nom `r@` suivi de son argument. La seconde ligne de l'exemple permet d'afficher la valeur de cette macro interne dans le terminal⁹⁴.

Exemple 56

	code	résultat
1	<code>\label{frame:csname}</code>	
2	<code>\expandafter\meaning\csname r@frame:csname\endcsname</code>	
		<code>macro:->{}{78}</code>

Une référence interne est ainsi créée sous la forme d'une macro dont la définition est constituée de deux groupes contenant le numéro de l'élément (ici vide, car non numéroté) et le numéro de la page où il se trouve. La commande `\ref` utilisera le premier groupe dans la valeur de la macro tandis que `\pageref` utilisera le second⁹⁵. On trouvera plus d'informations à ce sujet, ainsi que de nombreux exemples, dans les actes de la conférence prononcée par Amy Hendrickson lors du TUG 2012 [5].

`expl3` reprend ce concept pour le type d'argument `c – c` comme `csname`.

Exemple 57

1	<code>\dim_set:Nn \l_tmpa_dim { 1cm }</code>
2	<code>\dim_set:cn { \l_tmpb_dim } { 2cm }</code>
3	<code>\dim_set_eq:cc { \l_tmpa_dim } { \l_tmpb_dim }</code>

Contrôle d'expansion

L'expansion désigne le remplacement d'une macro par sa signification. En s'en tenant aux commandes classiques de `\LaTeX`, le résultat de l'expansion d'une macro peut être affiché en utilisant les commandes suivantes (la première affiche dans le document, la seconde dans le terminal) :

94. NdT : nous développons ici le texte original pour plus de clarté.

95. Voir note 94 ci-dessus.

```

1 \meaning\commande
2 \show\commande

```

On a ainsi :

Exemple 59

```

1 \newcommand*\{\variable}{def}
2 \meaning\variable\
3 \newcommand*\{\fonction}[1]{fonction avec argument (#1)}
4   }
\meaning\fonction

```

macro:->def
macro:#1->fonction avec argument (#1)

code

résultat

La macro `\variable` est donc simplement remplacée par la chaîne de caractères `def`.

La macro `\fonction` accepte un argument et le place entre les parenthèses au lieu du `#1`.

Pour mettre en œuvre des structures plus complexes, d'autres macros sont souvent utilisées à l'intérieur des définitions de macros. Une expansion en plusieurs étapes est alors nécessaire.

Expansion en plusieurs étapes

Nous utilisons les définitions suivantes⁸⁶ à titre d'exemple pour illustrer la manière dont LATEX traite les commandes.

Exemple 60

```

1 \newcommand\ei{{eins}}
2 \newcommand\zei{{\ei,zwei}}
3 \newcommand\dei{{\ei,\zei,drei}}

```

Imaginons que chaque macro soit une boîte dont le contenu varie en fonction de sa définition :

\ei{{eins}}

Sans expansion, TeX ne voit qu'un *token*. Si l'on veut poursuivre le traitement, il faut que la boîte soit déballée :

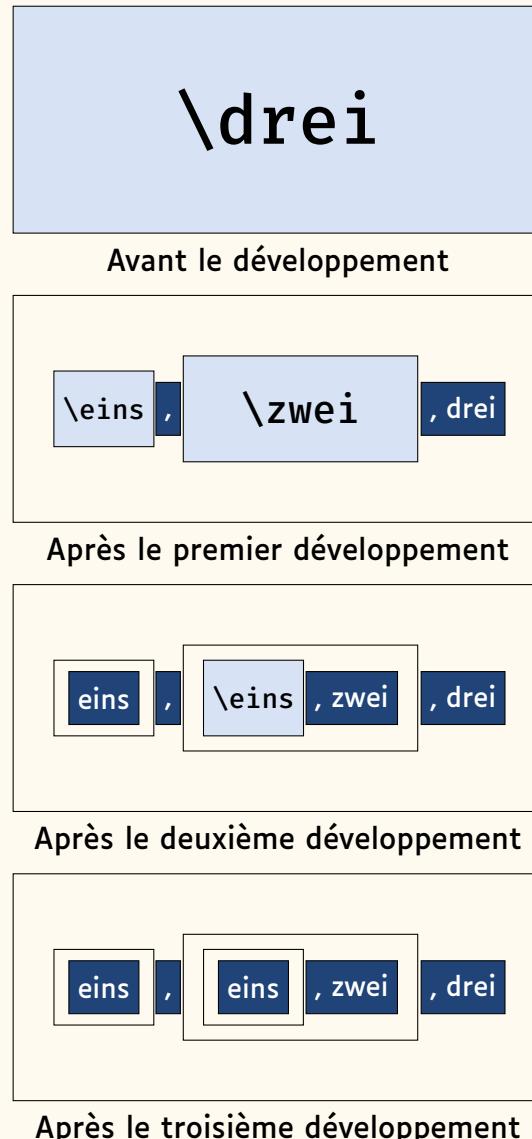
eins

Comme cette macro ne contient que la chaîne de caractères `eins`⁸⁶, ce processus ne peut pas être répété. La macro est déjà entièrement développée.

Si la macro `\ei{{eins}}` n'est développable qu'une fois, `\dei` peut être développée plusieurs fois. Les différentes étapes de développement sont illustrées

en figure 21.

FIGURE 21 – Illustration des étapes de développement dans des boîtes.



`expl3` permet, via les arguments, de contrôler explicitement – et très précisément – jusqu'à quel point les boîtes sont développées avant d'être traitées par la fonction.

On peut dès lors différencier les types d'arguments selon ce critère en utilisant directement une syntaxe `expl3`.

L'instruction `\tl_to_str:n` insère directement dans le document la chaîne de caractères qui représente son argument. Pour contrôler les niveaux de développement, on peut la faire précédé de `\exp_args:No` ou d'autres variantes de cette commande. Le premier argument de `\exp_args` est la macro dont on va traiter l'argument : il ne sera donc pas développé. Le deuxième argument de `\exp_args` est l'argument de la macro qui constitue son premier argument ; on en contrôle le développement grâce à la seconde lettre qui suit les deux points, comme le montrent les exemples suivants⁹⁶ :

96. Voir note 95 page 78.

Exemple 61

code

```

1 \ExplSyntaxOn
2 % sans développement
3 \tl_to_str:n { \drei }           \\
4 % développement simple
5 \exp_args:Nn \tl_to_str:n { \drei } \\
6 % développement total
7 \exp_args:Nf \tl_to_str:n { \drei } \\
8 % développement ininterrompu (e)
9 \exp_args:Ne \tl_to_str:n { \drei }
10 \ExplSyntaxOff

```

résultat

```

\drei
\ eins ,\ zwei ,\ drei
eins,\ zwei ,\ drei
eins,eins,zwei,drei

```

Comme on le voit, il est possible de contrôler précisément l'ordre dans lequel le développement des arguments des fonctions sera effectué. Il est ainsi possible de regarder à l'intérieur des boîtes avant qu'elles ne soient traitées. Ainsi, on peut par exemple vérifier dans quel format une date apparaîtra et, en fonction de cela, faire un choix définitif :

Exemple 62

```

1 \cs_new:Nn \__dantetut_parse_date:n {
2   % La chaîne de caractères est divisée
3   % suivant les traits d'union :
4   \seq_set_split:Nnn \l_tmpa_seq { - } {#1}
5   \int_compare:nTF {\seq_count:N \l_tmpa_seq > 1}
6   {
7     % Il y avait effectivement des traits d'union :
8     % on suppose que la date était au format ISO.
9     \seq_item:Nn \l_tmpa_seq { 3 } .
10    \seq_item:Nn \l_tmpa_seq { 2 } .
11    \seq_item:Nn \l_tmpa_seq { 1 }
12  }{
13    % D'autres possibilités existent.
14    % Dans ce cas, on suppose que la date
15    % est déjà mise en forme correctement.
16    #1
17  }
18

```

La macro vérifie que l'argument contient un trait d'union. Si c'est le cas, la date est interprétée comme une date ISO (AAAA-MM-JJ). Sinon, on suppose que la date a le format JJ.MM.AAAA. Il est bien sûr possible de le vérifier ensuite.

Pour l'exemple, imaginons maintenant que la date qui a été renseignée au

début du document via la commande `\date` passe dans notre macro. En interne, la valeur est stockée dans la macro `\@date`. L'argument passé à `__dantetut_parse_date` est alors le *token* `\@date`, et non une chaîne de caractères : pour pouvoir examiner le contenu de la date, il faut donc d'abord développer la macro⁹⁷.

Une possibilité serait d'utiliser la commande `\exp_args`. Mais `expl3` fournit un mécanisme permettant de créer des variantes d'une commande de base :

```
1 \cs_generate_variant:Nn \__dantetut_parse_date:n {x}
```

Maintenant qu'existe la commande `__dantetut_parse_date:x`⁹⁸, il est possible de rédiger le code suivant :

Exemple 64

code

```
1 \ExplSyntaxOn
2 \__dantetut_parse_date:n { 23.06.2022 }
3 =
4 \__dantetut_parse_date:n { 2022-06-23 } =
5 \__dantetut_parse_date:x {
6   \use:c { \@date } % ou \makeatletter\@date
7     \makeatother
8 }
```

`\ExplSyntaxOff`

résultat

23.06.2022=23.06.2022=23.06.2022

Résumé conclusif et perspectives

Nous avons vu que, malgré – ou grâce à – une syntaxe qui peut paraître un peu étrange au premier abord, `LATEX3` et `expl3` étendent et simplifient la création de macros, en les rendant plus facilement adaptables. En complément du type de données de base (et unique) token, des types structurés de données sont introduits et une distinction est faite entre le traitement et le stockage des données. En outre, `expl3` rend le contrôle du développement des macros beaucoup plus transparent qu'auparavant.

Il est ainsi possible d'automatiser le traitement de différents contenus et d'écrire des macros dont le comportement varie en fonction des arguments fournis et de leur structure. En outre, la structure uniforme des interfaces facilite énormément la lecture et la compréhension de codes écrits par d'autres⁹⁹.

De très nombreuses macros, qui ont été redéfinies à plusieurs reprises dans de nombreux packages, sont ainsi obsolètes, car il existe désormais des fonctions prêtes à l'emploi dans le noyau lui-même. Des familles entières de

97. Voir note 94 page 78 (promis, cette fois-ci pas de double renvoi!).

98. NdT : ... qui développe complètement son argument, selon le sens de x vu plus haut.

99. NdT : une fois familiarisé avec les conventions de la syntaxe!

packages sont déjà remplacées par des modules standards L^AT_EX3, comme par exemple les nombreux packages permettant d'implémenter le parsage des options du type clé/valeur. À long terme, cela réduira les conflits entre les packages, tout comme d'autres extensions du noyau, et rendra le système L^AT_EX encore un peu plus stable.

Cet article ne donne cependant qu'un très petit aperçu des possibilités. Celles-ci étaient certes déjà illimitées avec L^AT_EX classique, mais les fonctions avancées nécessitaient souvent des *patches* et des *hacks* de bas niveau. Le package `expl3` m'a permis beaucoup plus simplement de programmer directement en L^AT_EX.

J'espère vraiment que cet article permettra d'éclairer un peu le chemin qui nous mène vers `expl3`.

`\prg_do_nothing`: ou `\relax`

Marei Peischl

traduit par Patrick Bideault

Références

- [1] Johannes L. BRAAMS et Javier BEZOS. *Babel, Localization and internationalization*. English. 2022. URL : <http://mirrors.ctan.org/macros/latex/required/babel/base/babel.pdf>.
- [2] Joseph WRIGHT. *Registering expl3 module*. English. 2012. URL : <https://www.texdev.net/2012/11/04/registering-expl3-module/>.
- [3] The L^AT_EX PROJECT. *The L^AT_EX3 Interfaces*. English. 2022. URL : <http://mirrors.ctan.org/macros/latex/contrib/l3kernel/interface3.pdf>.
- [4] *The expl3 package and L^AT_EX3 programming*. English. 2022. URL : <http://mirrors.ctan.org/macros/latex/contrib/l3kernel/expl3.pdf>.
- [5] Amy HENDRICKSON. *TUG 2012 Conference Proceedings*. English. 2012. URL : <https://www.tug.org/tug2012/booklet/hendricksonAmyTugProc.pdf>.



♪ PAS À PAS : INSTALLATION, VIA LE RÉSEAU, DE LA TEX LIVE 2024

La TeX Live 2024 vient de sortir. Pour les plus adeptes des nouveautés de monde de TeX, c'est donc l'heure d'une installation toute fraîche. Notre cher Denis Bitouzé a produit un document très complet et très détaillé que vous pouvez consulter ici :

<https://dgxy.link/en-ligne0>

Denis détaille l'installation sur quasiment tous les systèmes. Pour les systèmes Unix, il préconise une installation à partir du téléchargement de l'image .iso. Ce n'est pas ma méthode d'installation préférée, je préfère la *net-install* pour laquelle on ne télécharge que le script perl de lancement d'installation. À la suite de son exécution, le contenu de la TeX Live est téléchargé extension par extension : cette installation nécessite donc une connexion à internet stable.

Je vais décrire ici, pas à pas, l'installation que j'ai effectuée sous GNU/Linux ainsi que sous MacOSX. Cette installation se fait en ligne de commande dans un terminal. Cela peut être un peu effrayant, mais une fois que l'on a pris le pli, c'est d'une efficacité redoutable. Je vais traiter le cas simple d'une installation complète où l'on conserve la plupart des valeurs par défaut. Si vous souhaitez personnaliser davantage votre installation (par exemple n'installer qu'une sélection d'extensions), vous trouverez les informations nécessaires dans le guide complet traduit en français à l'adresse suivante : <https://tug.org/texlive/doc/texlive-fr/texlive-fr.html>

Téléchargement et préparation

Il faut commencer par télécharger l'archive .tar.gz. Pour cela, on va se placer dans un répertoire de travail temporaire, par exemple /tmp, et récupérer l'archive avec la commande wget (mais cela pourrait être fait aussi avec curl).

```
$ cd /tmp
```

```
/tmp $ wget  
/tmp $ https://mirror.ctan.org/systems/texlive/tlnet  
/install-tl-unx.tar.gz
```

Une fois cette archive téléchargée, nous allons l'extraire avec la commande suivante :

```
/tmp $ zcat < install-tl-unx.tar.gz | tar xf -
```

Il suffira ensuite de se déplacer dans le répertoire ainsi créé : son nom commence par `install-tl` suivi d'une date (le jour de la rédaction de cet article, il s'agit de `install-tl-20240326`). Pour se déplacer dans le

répertoire sans saisir la date de la TeX Live, on peut utiliser la commande suivante :

```
/tmp $ cd install-tl-*
```

Ça y est : nous sommes prêts à lancer l'installation.

Installation

Nous présentons une installation en root pour que cette installation soit accessible à tous les utilisateurs et utilisatrices de la machine.

Pour lancer l'installation, il suffit de lancer la commande suivante :

```
/tmp/install-tl-20240326 sudo perl install-tl  
Loading https://mirror.ibcp.fr/pub/CTAN/systems/texlive  
/tlnet/tlpkg/texlive.tlpdb  
Installing TeX Live 2024 from: https://mirror.ibcp.fr/  
pub/CTAN/systems/texlive/tlnet (not verified)  
Platform: universal-darwin => 'MacOSX current (10.14-)  
on ARM/x86_64'  
Distribution: net (downloading)  
Using URL: https://mirror.ibcp.fr/pub/CTAN/systems/  
texlive/tlnet  
Directory for temporary files: /tmp/dqNvdqMuYz
```

```
An existing installation of TeX Live has been found in  
/usr/local/texlive/2023
```

```
If you want its selection of schemes/collections and  
various options  
to be used, press `y', otherwise anything else.
```

```
Import settings from previous TeX Live installation: (y  
/n):
```

Dans mon cas, la TeX Live 2023 étant déjà installée, l'installateur de la version 2024 le détecte et me propose d'importer les réglages opérés pour l'installation de la version 2023, ce à quoi j'ai répondu oui en tapant y puis sur la touche ↵.

Le terminal affiche alors les lignes suivantes :

```
=====> TeX Live installation procedure <=====  
Letters/digits in <angle brackets> indicate  
menu items for actions or customizations  
help=> https://tug.org/texlive/doc/install-tl.html
```

```
Detected platform: MacOSX current (10.14-) on ARM/  
x86_64
```

```
<B> set binary platforms: 1 out of 15  
  
<S> set installation scheme: scheme-custom  
  
<C> set installation collections:  
    40 collections out of 41, disk space required:  
    8647 MB (free: 816603 MB)  
  
<D> set directories:  
    TEXDIR (the main TeX directory):  
        /usr/local/texlive/2024  
    TEXMFLOCAL (directory for site-wide local files):  
        /usr/local/texlive/texmf-local  
    TEXMFSYSVAR (directory for variable and  
    automatically generated data):  
        /usr/local/texlive/2024/texmf-var  
    TEXMFSYSCONFIG (directory for local config):  
        /usr/local/texlive/2024/texmf-config  
    TEXMFVAR (personal directory for variable and  
    automatically generated data):  
        ~/Library/texlive/2024/texmf-var  
    TEXMFCONFIG (personal directory for local config):  
        ~/Library/texlive/2024/texmf-config  
    TEXMFHOME (directory for user-specific files):  
        ~/Library/texmf  
  
<O> options:  
    [ ] use letter size instead of A4 by default  
    [X] allow execution of restricted list of programs  
        via \write18  
    [X] create all format files  
    [X] install macro/font doc tree  
    [X] install macro/font source tree  
    [ ] create symlinks to standard directories  
  
<V> set up for portable installation
```

Actions:

- <I> start installation to hard disk
- <P> save installation profile to 'texlive.profile'
and exit
- <Q> quit

Enter command:

Ici, pour que les programmes installés soient directement accessibles, je conseille de créer les liens symboliques dans les répertoires standards (*create symlinks to standard directories*). En effet, pour qu'on puisse avoir accès aux programmes sous Unix, il faut que les exécutables soient dans des répertoires que le système d'exploitation explore. Ici, on tapera 0 puis  pour configurer les options. Une fois cela fait, le prompt du terminal

affiche :

```
=====
Options customization:
```

```
<P> use letter size instead of A4 by default: [ ]
<E> execution of restricted list of programs: [X]
<F> create all format files: [X]
<D> install font/macro doc tree: [X]
<S> install font/macro source tree: [X]
<L> create symlinks in standard directories: [ ]
      binaries to:
      manpages to:
      info to:
```

Actions: (disk space required: 8647 MB)

```
<R> return to main menu
<Q> quit
```

Enter command:

À cette demande, on répondra par L suivi de . Le programme d'installation nous demandera alors d'entrer de nouvelles valeurs pour le répertoire des binaires, celui de la documentation et celui des informations. Je conseille de laisser les valeurs par défaut et donc de laisser la réponse vide en tapant simplement trois fois sur .

```
New value for binary directory [/usr/local/bin]:
New value for man directory      [/usr/local/man]:
New value for info directory     [/usr/local/info]:
```

Le terminal affichera ensuite :

```
=====
Options customization:
```

```
<P> use letter size instead of A4 by default: [ ]
<E> execution of restricted list of programs: [X]
<F> create all format files: [X]
<D> install font/macro doc tree: [X]
<S> install font/macro source tree: [X]
<L> create symlinks in standard directories: [X]
      binaries to: /usr/local/bin
      manpages to: /usr/local/man
      info to: /usr/local/info
```

Actions: (disk space required: 8647 MB)

```
<R> return to main menu
<Q> quit
```

Enter command:

Ce à quoi, puisque la configuration est faite, on retournera au menu principal en répondant R puis ↵.

Nous sommes alors prêts à lancer l'installation en répondant I puis ↵ à la demande d'action suivante :

```
...
<0> options:
[ ] use letter size instead of A4 by default
[X] allow execution of restricted list of programs via
    \write18
[X] create all format files
[X] install macro/font doc tree
[X] install macro/font source tree
[X] create symlinks to standard directories

<V> set up for portable installation

Actions:
<I> start installation to hard disk
<P> save installation profile to 'texlive.profile' and
    exit
<Q> quit

Enter command: I
```

Une fois cela fait, l'installation débutera et vous devriez alors voir quelque chose comme :

```
Installing to: /usr/local/texlive/2024
Installing [1/4, time/total: ?:?:?/??:?]: hyphen-base
[23k]
Installing [2/4, time/total: 00:00/00:00]: kpathsea
[1065k]
Installing [3/4, time/total: 00:00/00:00]: texlive-
    scripts [537k]
Installing [4/4, time/total: 00:01/00:01]: texlive.
    infra [557k]
Time used for installing the packages: 00:01
Installing [0001/4680, time/total: ?:?:?/??:?]: 12many
[376k]
Installing [0002/4680, time/total: 00:00/00:00]: 2up
[56k]
Installing [0003/4680, time/total: 00:00/00:00]:
    a0poster [119k]
Installing [0004/4680, time/total: 00:00/00:00]: a2ping
[69k]
Installing [0005/4680, time/total: 00:01/01:59:51]:
    a2ping.universal-darwin [1k]
Installing [0006/4680, time/total: 00:01/01:59:47]:
    a4wide [133k]
Installing [0007/4680, time/total: 00:01/01:38:38]:
```

```
a5comb [91k]
Installing [0008/4680, time/total: 00:01/01:28:01]:
  aaai-named [6k]
Installing [0009/4680, time/total: 00:01/01:27:30]:
  aalok [187k]
Installing [0010/4680, time/total: 00:01/01:11:43]:
  aastex [1188k]
Installing [0011/4680, time/total: 00:02/01:06:41]:
  abbr [4k]
Installing [0012/4680, time/total: 00:02/01:06:35]: abc
  [288k]
Installing [0013/4680, time/total: 00:02/58:58]: abnt
  [121k]
Installing [0014/4680, time/total: 00:02/56:16]:
  abntex2 [4508k]
Installing [0015/4680, time/total: 00:04/41:29]:
  abnntexto [207k]
Installing [0016/4680, time/total: 00:04/40:19]:
  aboensis [34288k]
```

Une fois l'installation terminée, vous aurez, à la fin, le message suivant :

```
Bienvenue sous TeX Live !
Voir /usr/local/texlive/2024/index.html pour les liens
vers la documentation.
```

```
Le site web TeX Live (https://tug.org/texlive/) fournit
toutes les mises à jour et corrections.
TeX Live est un projet commun des groupes d'utilisa-
teurs de TeX dans le monde entier ;
veuillez envisager de le soutenir en rejoignant le
groupe qui vous convient le mieux.
La liste des groupes est disponible sur le Web à l'ad-
resse https://tug.org/usergroups.html.
```

Logfile: /usr/local/texlive/2024/install-tl.log

Ça y est, l'installation est terminée, et vous pouvez compiler vos documents avec les programmes de la dernière TeX Live!

N'hésitez pas à mettre à jour régulièrement votre distribution pour avoir accès aux nouveautés, notamment documentées dans la bonne *vieille veille* TeXnologique de cette *Lettre*, avec la commande suivante :

```
$ sudo tlmgr update --all
```

Bonnes compilations!

Maxime Chupin



⌚ FEUILLE ALDINE OU... GRAPPE ?

Les lecteurs de cette *Lettre GUTenberg* ont sûrement remarqué que les titres de section (comme celui de cette note !) y sont marqués d'une splendide ⌚ (qu'on appelle souvent « feuille aldine », doublement à tort comme nous le verrons) reprenant en cela le style des *Cahiers* défini par Michel Bovani et Thierry Bouche depuis le n° 46-47 en 2006. Ces *Cahiers* et *Lettre GUTenberg* utilisent un « ornement » de la fonte Fourier dessinée par Michel Bovani¹⁰⁰. Cette fonte définit quatre « feuilles aldines », répertoriées dans le tableau suivant :

Commande	Glyphe
\aldineleft	⌚
\aldineright	⌚
\aldine	⌚
\aldinesmall	⌚

Mais, sait-on bien ce que sont ces signes ? On va en raconter brièvement l'histoire et l'usage, tout en insistant sur... leurs noms.

Ces signes sont souvent classés avec ce que l'on appelle *vignettes* (du nom des pampres de vignes souvent utilisées par les miniaturistes dans les manuscrits médiévaux), *arabesques* (du fait de leurs origines), *fleurons* (allusion aux décors en forme de fleurs mais terme que l'on garde surtout pour les vignettes de Granjon et de son époque), voire, de façon plus générique, *ornements*. Ces ornements ont attiré l'attention des historiens de la typographie, dont notamment AUDIN [1], MEYNELL et MORISON [2], DREYFUS [3] et plus récemment CAFLISCH [4] et surtout VERVLIET [5, 6] et KEMP [7]. Toutefois, ces études concernent essentiellement les aspects graphiques (et quelques fois typologiques) de ces signes, beaucoup plus rarement leurs usages.

Dès l'antiquité grecque puis romaine, des motifs ornementaux utilisent des végétaux ; ils se retrouvent dans l'art byzantin, puis médiéval et islamique. Le livre manuscrit en utilise abondamment encore au xv^e siècle et ce style perdure avec les premiers incunables où la décoration était faite à la main sur les feuilles du texte imprimé (figure 22 page suivante). Le besoin de combler les lignes creuses (comme pour le livre médiéval) est probablement à l'origine de l'utilisation de types de remplissage formés de motifs floraux (figure 23 page ci-contre : *Ars moriendi*, 1478). Robert Granjon dessina, vers 1540, les plus célèbres de ces *fleurons* et même des bandeaux entiers (VERVLIET [6]) qui, par combinaisons [8], permettaient de créer des ornements géométriques. Ceux-ci seront très utilisés dans les bandeaux des textes imprimés jusqu'au xix^e siècle (comme celui de Fertel en page suivante). Par la suite, ils se modernisèrent avec les figures, cadres et coins de Fournier, puis ceux apparaissant avec l'Art nouveau, etc.

Mais les signes qui nous concernent ici, comme la feuille aldine, ⌚, et quelques rares autres (notamment le « gland de Granjon, ⌚ »¹⁰¹), sont à

100. Depuis, cette fonte et les packages correspondants ont été adaptés au format OpenType avec la collaboration de Daniel Flipo. L'ensemble est donc utilisable avec LATEX, LualATEX ou XETEX. Voir <https://www.ctan.org/pkg/fourier>.

101. ⌚ est obtenu ici en Fourier par la commande \decotwo.

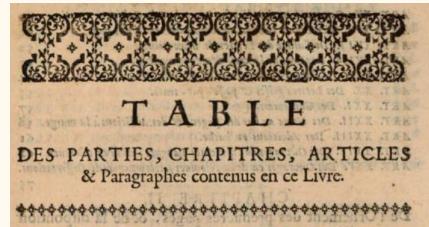
FIGURE 22 – Gasperini epistolae, premier livre imprimé en France (presses de la Sorbonne, 1471). [Gallica]



Source gallica.bnf.fr / Bibliothèque nationale de France

FIGURE 23 – À gauche : Giovanni e Alberto Alvise, *Ars Moriendi*, Vérone, 1478. À droite : Extrait de M.D. Ferret, *La Science pratique de l'imprimerie*, 1723.

Questo adócha tractato utilissimo cõtene sei particelle de le quale la principale sequita. La prima sera de le laude de la morte e de la fietia de la pere larte de benmoriore. La secuda e de letemptatione diabolice le quale



mettre à part car :

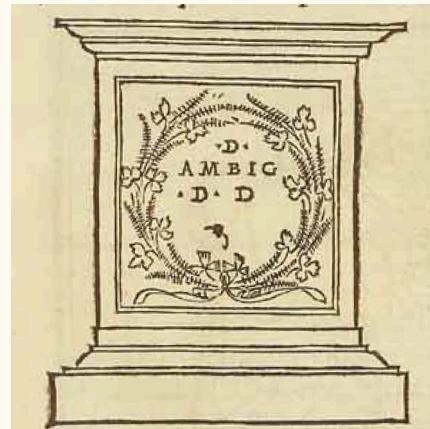
1. ils s'emploient en général isolés (leur dessin ne se prête pas à des combinaisons géométriques),
2. leur glyphe est très noir, comme une puce •,
3. ils n'ont pas qu'un rôle purement décoratif.

On va voir en effet qu'ils avaient un rôle de ponctuation, de marqueur syntaxique ou d'accroche.

Dès l'antiquité grecque, la séparation des mots puis la structuration du texte devinrent une nécessité : l'usage de caractères autres que les lettres s'imposa et ainsi apparurent les signes de ponctuation [9, 10] et autres marqueurs dont les « fleurons » utilisés à Palmyre vers 100 av. J.C. pour préfixer les nombres (nous y reviendrons plus bas) et la feuille de lierre¹⁰² (en latin *hedera*, c'est le terme générique utilisé internationalement en épigraphie) et sa variante stylisée en forme de triangle. Ces dernières servent de caractère de remplissage, de point de phrase, mais aussi d'accroche. Par exemple, dans une inscription à Rome (page suivante, à gauche), on peut voir deux feuilles de lierre (bien en forme de cœur !); l'une, ligne 4, qui sert de remplissage, l'autre, entre D et M (*Dis Manibus*, aux dieux mânes), en revanche sert à attirer l'œil, à marquer le début de l'épitaphe.

102. La feuille du lierre a une forme de cœur graphique. Le lierre étant le symbole de la longévité, son image (appelée cœur) serait devenue le symbole de l'amour.

FIGURE 24 – À gauche : Inscription latine du II-III^e siècle. [American Academy in Rome]. À droite : Dessin d'inscription latine (avec une feuille aldine) du *Songe de Poliphile*, imprimé par Alde Manuce à Venise, 1499.



La naissance du livre imprimé a lieu en pleine Renaissance, période où on est très curieux des antiquités. Dès que l'on a su imprimer des images (fin XV^e siècle), paraissent des livres avec des représentations de ces inscriptions romaines. C'est ainsi que dans le *Songe de Poliphile* de Colonna, imprimé à Venise par Alde Manuce, on voit (de la présente page, à droite) une colonne romaine avec une feuille aldine¹⁰³.

En Allemagne, la ville de Augsburg faisait concurrence à Venise et était le fief de nombreux artistes et imprimeurs. C'est là que paraît en 1505 un livre intitulé *Romanae vetustatis fragmēta* par Konrad Peutinger, imprimé par Erhard Ratdolt, comprenant plusieurs pages décrivant des inscriptions latines ; celles-ci ont été composées avec des caractères gravés par Ratdolt. Et parmi eux se trouvent des signes de ponctuation : triangles et feuilles de vigne (figure 25).

FIGURE 25 – Extraits de *Romanae vetustatis fragmēta* imprimés par Erhard Ratdolt, 1505.

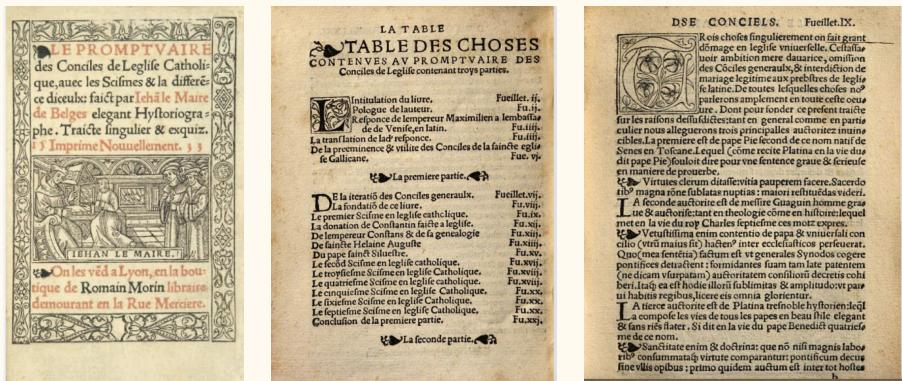


Ce nouveau signe semble avoir été vite adopté et plusieurs graveurs en offrent, tant dans la vallée du Rhin qu'en Italie et, dès 1515, en France à Lyon puis Paris (Colines, Garamont, Granjon, etc.), à Anvers, etc. On est alors à une cinquantaine d'années après l'invention de l'imprimerie et on cherche toujours à remplacer la couleur des manuscrits médiévaux par quelque chose qui permette (entre autres) de montrer la structure logique du texte. Ce à quoi Gutenberg n'est pas arrivé car ses impressions nécessitaient le passage de rubricateurs qui faisaient à la main sur chaque exemplaire imprimé ce qu'ils auraient fait sur un manuscrit. Apparaît alors la page de titre, concept de la toute fin du XV^e siècle et objet de recherches et innovations de la part des imprimeurs [11]. Puis les éléments structurants (tête de chapitre, de section, alinéas, débuts de phrase, etc.) commencent à être composés avec des attributs spécifiques (pour les distinguer), comme

103. Profitons en pour dire qu'il s'agit ici d'un dessin gravé, et que Alde Manuce n'a jamais utilisé une seule de ces feuilles dans ses compositions typographiques. L'expression « feuille aldine » est donc erronée, sauf si aldine fait allusion aux successeurs de Manuce vers 1550 qui, eux, en utilisèrent beaucoup !

l'emploi de capitales, parfois des changements de corps, voire quelques rares signes spéciaux comme le pied-de-mouche (¶ ou ¶¹⁰⁴). Bien sûr pas de gras (inventé au xix^e siècle) et même l'italique ne servait pas encore à la distinction. Parmi les nombreux livres qui utilisèrent le signe §, regardons *Le Promptuaire des conciles de l'Eglise catholique...* (de Jean Lemaire de Belges, imprimé à Lyon par Romain Morin en 1532) dont trois pages sont présentées en figure 26.

FIGURE 26 – Trois pages du *Promptuaire des conciles de l'Eglise catholique...* de Jean Lemaire de Belges, imprimé à Lyon par Romain Morin en 1532.



De gauche à droite de la figure 26, la page de titre montre (outre une gravure sur bois encadrée d'entrelacs et figures géométriques) deux cartouches « préfixées » de ¶ (jouant ainsi le rôle de notre \bfseries); la table des matières dont le titre est également préfixé de ¶ et dont les sous-titres sont entourés (parenthésés) de ¶ symétriques; enfin une page de texte où ¶ indique des débuts d'alinéas. Ce sont les principaux usages qui vont être faits de ce « nouveau » signe, auxquels il convient d'ajouter un rôle plus décoratif : terminer un cul-de-lampe triangulaire et remplir le blanc entre deux titres (notamment dans la page de couverture).

Énormément de livres de la première moitié du xvi^e siècle sont alors composés ainsi avec ce signe, dont voici quelques exemples typiques : en figure 27 page suivante, l'aldine est utilisée pour préfixer des titres tandis que deux aldines permettent de parentheser du texte en figure 28 page suivante ; enfin, en 29, l'aldine joue un rôle décoratif.

Dans une étude quasi-exhaustive de la feuille de vigne, VERVLIET [5] en inventorie en fait deux types : celles ressemblant à un cœur (¶) que nous appelons donc ici « feuille aldine », et celles ressemblant davantage à une feuille (¶) que nous appelons « feuille de vigne ». Il me semble (en consultant nombre d'ouvrages des « apparences » citées par VERVLIET [5]) que ce sont essentiellement les aldines qui servirent à marquer les grands titres, sommaires, etc. tandis que ces dernières feuilles de vigne, du fait que plus tardives et moins noires, jouèrent surtout un rôle de décoration (séparation de titres, bas de cul-de-lampe) et non de marqueur (figure 30 page 95). En revanche le gland de Granjon (inventorié par VERVLIET [12, fig. 17, 40-42]) semble avoir été utilisé comme notre feuille aldine. Notons que, sans doute de façon générique, Vervliet appelle *vine-leaf* (feuille de vigne) les deux signes ¶ et ¶ mais *flower* (fleur) le gland ¶ !

104. Fourier propose divers pieds-de-mouche : \oldpilcrowone=>¶, \oldpilcrowfive=>¶, etc.

FIGURE 27 – Titres préfixés. De gauche à droite : 1) Crispus, G.S., *Bella Duo, Catilinarium et Iugurthinum*, 1521 (<https://books.google.fr>). 2) J. Quinquareboreus, *De re grammaticæ Hebraeorum opus*, Paris, 1556 (feuille aldine de Garamont). 3) Rabelais, *Tiers livre des faictz et dictz...Pantagruel*, Toulouse, 1546 (feuille aldine de Fournier).

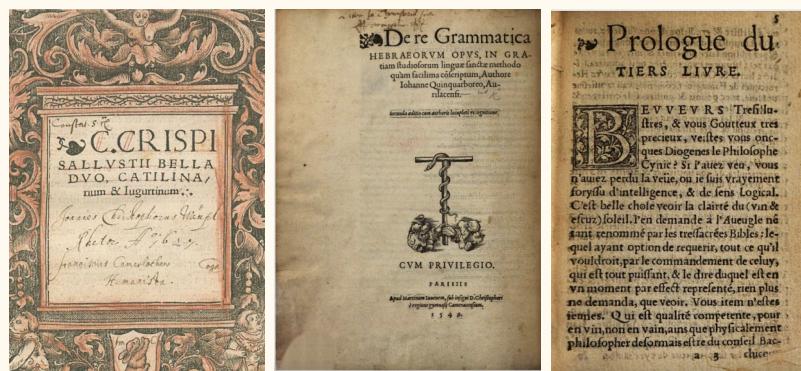


FIGURE 28 – Titres parenthésés. De gauche à droite : 1) Marguerite d'Angoulême, *Le Miroir de l'ame pecheresse*, Sedan, 1533. 2) Dupré et Collines, 1531. 3) Xenophon, *Las Obras*, Salamanques (feuille de Juan de Junta). 4) Jean de Meung, *Le Rommant de la Rose...*, Paris, Pierre Vidoue pour Gallois du Pré, 1523. 5) Caractères de Granjon, 1570 dans *Spécimen de Le Bé* (1599).

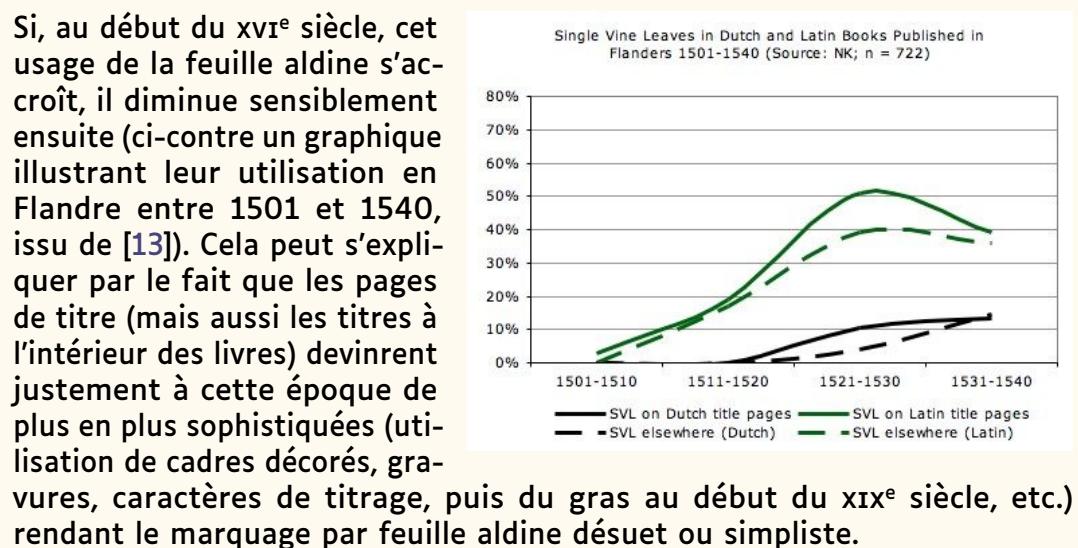
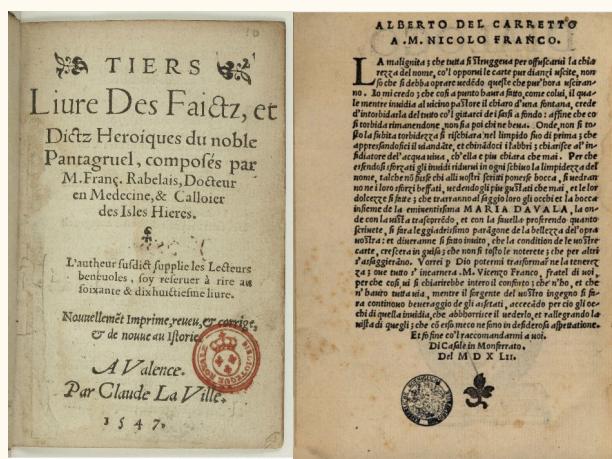


FIGURE 29 – Remplissage décoratif. De gauche à droite : 1) Konrad Peutinger, *Dédicace*, 1520. 2) Rabelais, *La vie ...du grand Gargantua...*, Lyon, 1542. 3) Marguerites de la marguerite, imprimé à Lyon par Jean de Tournes, 1547.



FIGURE 30 – Utilisation de « feuilles de vigne ». De gauche à droite : 1) Rabelais, *Tiers livre... de Pantagruel*, LaVille, Valence, 1547 (feuille aldine de Gryphius). 2) Niccolò Franco, *Dialogo*, Guidone, 1542. 3) Georgius Trapezuntius, *Dialectica*, Paris, Colines, 1528.



La feuille aldine ne devint plus qu'un simple signe décoratif mais, n'ayant pas les propriétés géométriques des fleurons, elle fut peu à peu abandonnée (en tant que signe isolé). On la trouve toutefois encore ici ou là, notamment dans les spécimens de caractères de Egenolff-Berner (1592), Lamesle (1742), Delacolonge (1773) (voir figure 31), etc. mais plutôt dans l'esprit « vignettes pour frises » (elles ne sont alors plus représentées comme signe isolé mais par une répétition du même glyphe).

FIGURE 31 – Spécimens de Lamesle, 1742 et de Delacolonge, 1773.

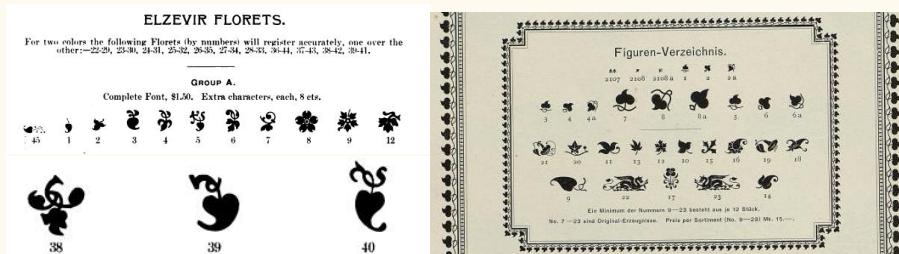


Mais ni Fertel (*La science pratique de l'imprimerie*, 1723), ni Fournier (*Manuel typographique*, 1764) n'y font allusion et elle semble bien être absente des casses aux XVIII et XIX^{es} siècles sauf, parfois, comme élément de frise. On ne voit aucune feuille aldine dans le spécimen de Gillé (1808), pourtant riche en fleurons et vignettes, ni dans les spécimens de ses successeurs

Balzac et Deberny (1828), ni dans les spécimens de l'Imprimerie royale puis nationale, ni chez Tarbé (le grand imprimeur parisien du XIX^e siècle).

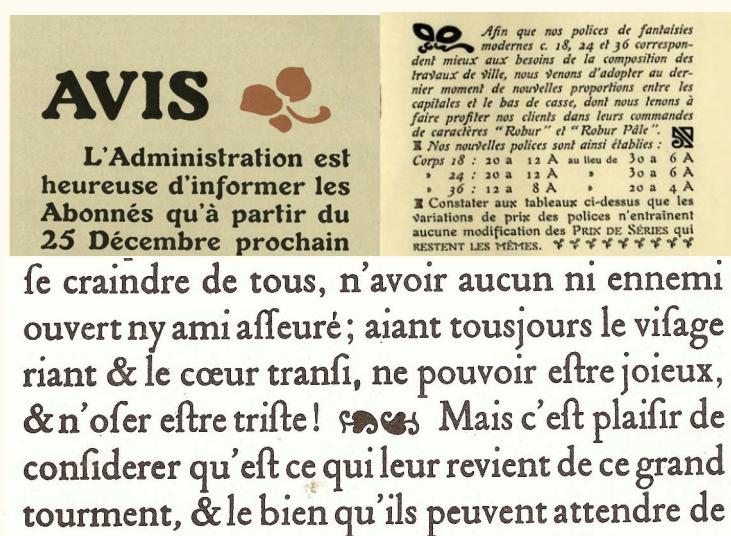
Pour diverses raisons (réaction à la mécanisation et au modernisme des linéales, retour à un certain humanisme, etc.), autour de 1840, les Anglais redécouvrent les caractères de Caslon et un renouveau du livre paraît grâce à la Chiswick Press et à William Morris. De même aux USA, en France (avec les elzévirs), en Allemagne, etc. Les vieux caractères d'ornement sont alors regravés et réapparaissent dans les spécimens (cette fois en tant que signes isolés, voir figure 32).

FIGURE 32 – À gauche : Cleveland type foundry, USA, 1893. À droite : Fonderie Kreb, Allemagne, 1899.



Le début du XX^e siècle confirme le renouveau des vieux caractères, y compris des vignettes et fleurons où Linotype puis Monotype excellent (ces fleurons sont alors l'objet de fascicules de démonstration par WARDE [14] puis CAFLISCH [15]). De son côté, Peignot, tout en re-dessinant les caractères anciens, crée de nouvelles vignettes, de style Art nouveau notamment. Il est intéressant de signaler que ces vignettes sont parfois utilisées seules, comme accroche, bouche-trou, ponctuation (exemple : [4]), etc., comme elles l'étaient au XVI^e siècle.

FIGURE 33 – En haut : Vignettes d'accroche et de remplissage ; Peignot, *Créations françaises en typographie moderne*, 1907. En bas : Justification par remplissage de feuilles aldinées, *Contr'un d'Étienne de la Boétie*, imprimé par Louis Jou, 1917.



Depuis 1950, ces fontes « plomb » ont été adaptées pour la photocomposition puis l'informatique et aujourd'hui les fontes d'ornements sont légion, parfois belles et fidèles¹⁰⁵, ou bien créatives...ou encore d'un goût douteux !

105. C'est notamment le cas de la série *Arabesques Ornements* de Monotype dont la troisième ne comprend que des fleurons isolés dont ♀ ♀ ♀.

La taille des fontes numériques (souvent une centaine de caractères seulement) a rendu nécessaire de créer de petits ensembles de ces ornements (et d'y ajouter des symboles « divers » comme les ciseaux, des flèches, étoiles, etc.). C'est ainsi que sont nées des fontes connues sous le nom de dingbats¹⁰⁶ dont surtout celles de Hermann Zapf (Zapf Dingbats dessinée en 1978 et incluse dans les fontes PostScript de l'Apple LaserWriter Plus en 1986) puis de Bigelow & Holmes (fontes Lucida Icons diffusées par Microsoft en 1992 sous le nom de Wingdings¹⁰⁷). Ces dingbats ont été normalisés par leur adoption par Unicode dès la version 1.1 (1993). À ceux de la première série, il en a été ajouté quelques-uns¹⁰⁸ depuis. Aujourd'hui on y trouve donc (sans mentionner les émoticônes) :

Glyphe	Numéro de code	NOMS ANGLAIS ET FRANÇAIS
❧	2619	REVERSED ROTATED FLORAL HEART BULLET CŒUR FLORAL COUCHÉ À DROITE
❧	2766	FLORAL HEART CŒUR FLORAL
❧	2767	ROTATED FLORAL HEART BULLET CŒUR FLORAL COUCHÉ
☙	10877	PALMYRENE LEFT-POINTING FLEURON FLEURON PALMYRÉEN POINTANT VERS LA GAUCHE
☙	10878	PALMYRENE RIGHT-POINTING FLEURON FLEURON PALMYRÉEN POINTANT VERS LA DROITE
☙	10AF1	MANICHAEAN PUNCTUATION FLEURON PONCTUATION MANICHIÉENNE FLEURON

Bien sûr, TeX n'a pas attendu Unicode pour manipuler des ornements, aussi trouve-t-on de nombreux paquets autres que Fourier proposant ces signes¹⁰⁹. Exemple :

Exemple 65

```
1 % Avec \usepackage{adforn,pifont,utfsym} en préambule
2 \adfhangingleafleft \ding{166} \usym{2767}
```

code

résultat



106. Le mot anglais *dingbat*, composé de *ding* (« sonner ») et de *bat* (« bâton »), signifierait idiot, imbécile, truc ou machin. En typographie on appelait *casseau* les casses où l'on mettait les caractères spéciaux pour un travail, les ornements étant parfois appelés « bouche-trou » du fait de leur usage, notamment dans la presse.

107. Wingdings n'a pas notre feuille aldine !

108. Ce sont les trois derniers du tableau ci-après. Ce sont des ponctuations relevées à Palmyre sur des inscriptions lapidaires d'environ 100 av. J.C. (voir Everson, <https://www.unicode.org/L2/L2010/10255r2-n3867.pdf>, page 8 et suivantes); le dernier est un préfixe de nombres. Alors que les trois premiers caractères sont fréquents dans les fontes Unicode – ici on utilise EB Garamond qui est dans TeXLive – ceux palmyréens sont difficiles à trouver. La fonte employée ici pour ces trois signes est unifont_upper.ttf de Gnu.

109. Voir notamment *The Comprehensive LATEX Symbol List* [16].

On remarquera qu'Unicode parle de « cœur » là où Fourier et adforn parlent de « fleur ». On voit dans la littérature diverses précisions pour ces feuilles ou fleurs, correspondant souvent au glyphe concerné : outre la vigne (*vine leaf*), citons le lierre (en latin *hedera*, en anglais *ivy leaf*), le trèfle *trefoil*, sans oublier le gland, etc.

Mais ce « cœur floral » intrigue. Effectivement, le glyphe peut ressembler à un cœur (ce qui était le cas déjà pour les feuilles de lierre chez les latins), mais... a-t-on bien regardé un sarment de vigne comme ci-contre ? On voit que la feuille de vigne est découpée, presque trifoliée, alors que la grappe est beaucoup plus compacte, foncée, parfois avec la pointe relevée — ça se voit surtout sur les gros raisins italiens. Alors, et si cette « feuille » aldine était une « grappe » ? C'est bien ce que montre la figure 34 avec à gauche une « grappe » aldine, au centre, un dessin actuel représentant une grappe de raisin, et à droite une « feuille » de vigne de Nicolini, 1525. En figure 35, la « grappe » aldine de gauche correspond bien à la grappe de raisin de droite.

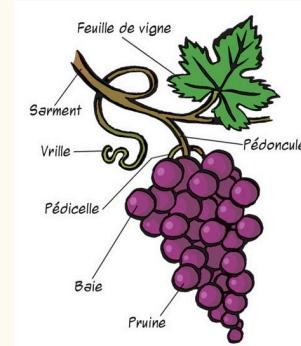
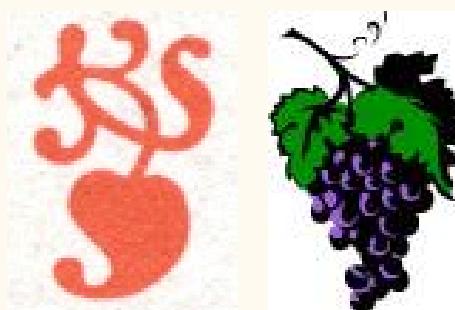


FIGURE 34 – Comparaison entre une feuille aldine (à gauche), un dessin de grappe de raisin (au centre), et une feuille de vigne (à droite).



FIGURE 35 – Comparaison entre une feuille aldine, à gauche, et un dessin de grappe de raisin, à droite.



Pour conclure

VERVLIET [5] a analysé deux signes différents qu'il réunit sous le nom de *vine leaf ornaments*. L'un ressemble à une feuille de vigne et l'autre à une grappe de raisin.

La feuille de vigne est dans la tradition des manuscrits médiévaux ; au xvi^e siècle, elle a servi, isolée, comme décoration (souvent pour séparer des titres) puis, par répétition, pour composer des frises — et ce jusqu'au xx^e siècle.

La grappe (on l'a appelée jusqu'ici « feuille aldine ») est issue plutôt de l'épigraphie latine (où elle avait la forme de cœur du lierre) et a été introduite brutalement en imprimerie par Ratdolt au début du xvi^e siècle (on propose

donc de l'appeler plutôt « grappe de Ratdolt »!). Pendant un siècle, elle a essentiellement servi de marqueur d'emphase (seule, en préfixe, ou par paire comme des parenthèses); occasionnellement elle a servi de décoration (bas des cul-de-lampes notamment). Du xv^e siècle au xx^e siècle, elle a subi à peu près le même sort que la feuille de vigne (emploi dans des frises).

C'est ce rôle d'accroche préfixée, datant du xv^e siècle, que l'on peut donc voir dans l'usage de la... grappe de Ratdolt de nos Cahiers et Lettre GUTenberg.

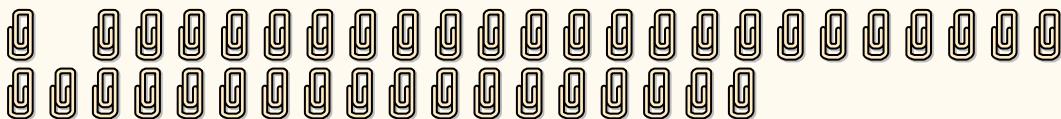
Jacques André



Références

- [1] Maurice AUDIN. « La vignette typographique ». In : *Arts et métiers graphiques* 2 (1927).
- [2] Francis MEYNELL et Stanley MORISON. « Printer's vine leafs and arabesques ». In : *The Fleuron, A Journal of Typography* 1 (1923).
- [3] John DREYFUS. « French type ornaments ». In : *Aspects of French eighteenth century typography*. Cambridge, 1982.
- [4] Max CAFLISCH. « Pour une typologie de la feuille aldine ». In : *Graphê* 30 (2006). v. aussi <http://jacques-andre.fr/japublis/aldine.pdf>.
- [5] Hendrik D.L. VERVLIET. *Vine Leaf Ornaments in Renaissance Typography*. Oak Knoll Press, 2012.
- [6] Hendrik D.L. VERVLIET. *Granjon's flowers*. Oak Knoll Press, 2016.
- [7] William KEMP. « Des fleurs typographiques à Venise, Lyon et Anvers ». In : *Bibliothèque d'Humanisme et Renaissance* 3 (2019), p. 533-542. URL : <https://www.jstor.org/stable/45282065>.
- [8] Yves PERROUSSEAUX et Jacques ANDRÉ. « Composer avec des vignettes à combinaison ». In : *Histoire de l'écriture typographique – le xviii^e siècle*. Gap : Adverbium/Atelier Perrousseaux éd., 2010, p. 174-207.
- [9] Paolo POCSETTI. « La réflexion autour de la ponctuation dans l'Antiquité gréco-latine ». In : *Langue française* 172 (2011), p. 19-35. URL : <https://www.cairn.info/revue-langue-francaise-2011-4-page-19.htm>.
- [10] M.B. PARKER. *Pause and Effect : Punctuation in the West*. Berkeley : University of California Press, 1993.
- [11] Jean-François GILMONT et Alexandre VANAUTGAERDEN. *La page de titre à la Renaissance. Treize études...* Brepols, 2008.
- [12] Hendrik D.L. VERVLIET. *Robert Granjon, letter-cutter*. Oak Knoll Press, 2018.
- [13] Goran PROOT. *The Single Vine Leaf, aka the "Aldine Leaf"*. 2013. URL : <https://www.folger.edu/blogs/collation/the-single-vine-leaf-aka-the-alanine-leaf/>.
- [14] Frederic WARDE. *Printers Ornaments applied to the composition of decorative borders, panels and patterns*. Londres : Lanston Monotype, 1928. URL : <http://jacques-andre.fr/faqtypo/orn/warde.pdf>.
- [15] Max CAFLISCH. *Kleines Spiel mit Ornamenten*. Zürich, 1965. URL : <http://jacques-andre.fr/faqtypo/orn/jeux-papier.pdf>.

- [16] Scott PAKIN. *The comprehensive L^AT_EX symbol list*. 2024. URL : <https://tug.ctan.org/info/symbols/comprehensive/symbols-a4.pdf>.



LA FONTE DE CE NUMÉRO : LUCIOLE

Pour ce numéro de la *Lettre*, nous mettons à l'honneur la fonte Luciole. Celle-ci a été conçue pour les personnes malvoyantes. Elle devrait donc intéresser des enseignants utilisateurs de T_EX.

Luciole est une fonte créée en France¹¹⁰ par une équipe pluridisciplinaire :

Coordination : Carole Malet et Jonathan Fabreguettes

Ophthalmologue : Dr^e Florence de Saint-Étienne

Orthoptiste : Anne-Céline Blanc

Psychologue : Véronique Morra

Dessinateur de caractères : Laurent Bourcellier

Transcriiteur : Jonathan Fabreguettes¹¹¹

Chercheurs : Dr^e Anna-Rita Galiano et Dr Nicolas Baltenneck

Étudiantes : Gaétane Hurstel, Camille Benas, Gaëlle Bonnesseur, Hind Drissi et Vanessa Augereau-Depoix

La composition de cette équipe montre que trois champs de compétences ont été réunis pour ce travail : le versant médical, le versant de l'édition spécialisée, et le versant typographique.

La démarche du projet est évidemment intéressante, le but étant en effet très noble. Il s'agit également d'un projet de recherche, et cette fonte, issue d'études, s'inscrit dans une démarche scientifique. En effet, l'adaptation aux personnes malvoyantes n'est pas simplement déclarative, mais repose bel et bien sur des critères précis, et se vérifie lors d'études scientifiques [1].

La problématique de la déficience visuelle est large. Par exemple, l'OMS¹¹⁴ distingue cinq catégories : déficience visuelle légère (acuité inférieure à 5/10), déficience visuelle modérée (inférieure à 3/10), déficience visuelle

110. Cocorico!

111. Nous constatons une fois encore que n'est pas mentionnée la personne qui a effectué le long travail de mise en cassetin, enfin : de remplissage de la table unicode (appelons-le « distribution », comme du temps du plomb). Peut-être cette fonction est-elle ici subsumée sous le terme générique de transcriuteur. Nous regrettons néanmoins qu'elle ne soit explicitement citée. Par exemple, lors de sa passionnante allocution à TypeParis¹¹², nous avons appris que David Březina effectua lui-même la distribution au sein de son important projet Rosetta¹¹³. Cette tâche très conséquente est complètement ignorée, alors qu'elle est d'une grande utilité. Nous ne comprenons pas pourquoi elle est passée sous silence.

112. <https://typeparis.com/events/david-brezina> – Le lectorat, toujours avisé, aura constaté que nous en avons profité pour placer un appel de note dans une note, ce qui est toujours un plaisir. Coupable, certes, mais plaisir quand-même... réalisable cependant à condition d'utiliser \footnotemark et \footnotetext, voir *Cahiers 12*!

113. <https://rosettatype.com/>

114. <https://icd.who.int/browse/2024-01/mms/en#1103667651>

sévère (inférieure à 1/10), cécité (inférieure à 1/20), déficience de vision de près. Avec ces critères, la déficience visuelle touche environ 1,7 million de lecteurs en France. En pratique, c'est principalement sur l'acuité visuelle de près que s'est concentrée l'équipe car c'est autour de cette problématique que les besoins en termes de lecture sont les plus grands.

Ce projet est très intéressant par bien des aspects. C'est tout d'abord un travail d'équipe. De plus, les auteurs, dans un souci de servir le plus et le mieux possible les malvoyants, ont choisi de diffuser la fonte Luciole gratuitement, avec une licence *libre connue* (*Creative Commons*). Nous ne pouvions qu'être sensibles à cet aspect ; c'est pourquoi nous avons choisi de présenter Luciole ici, même si elle n'est – pour l'instant – pas *packagée* pour *LAT_EX*.

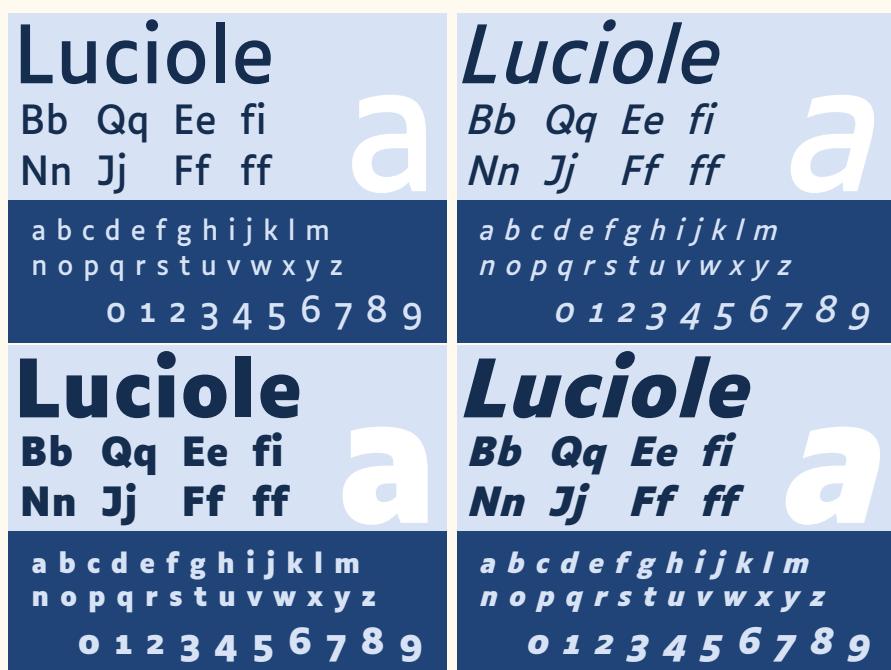
Dans cette présentation, nous nous sommes très largement inspirés de l'article « Méthode de conception d'un caractère typographique à l'usage des lecteurs déficients visuels » [2].

Design

La composition intégrale de ce numéro de la *Lettre* permet de bien se rendre compte du rendu de l'utilisation de la fonte Luciole pour un document complet. Cependant, comme à l'accoutumée désormais, nous présentons des échantillons montrant la fonte Luciole en figure 36.

Nous avons affaire ici à une fonte sans empattement ou *linéale*. Comme on le voit, le romain « *Regular* » est nettement plus gras que les autres linéales classiques (voir figure 38 page 104), et le « *bold* » est particulièrement très très gras¹¹⁵.

FIGURE 36 – La fonte Luciole : romain, italique, gras et gras-italique.



115. Dommage que Luciole n'utilise pas les concepts de fonte variable !

L'équipe s'est concentrée sur 12 critères de conception. Nous reproduisons ici l'illustration de certains de ces critères dans [2], avec `Luatex`, `fontspec` et quelques fois d'autres choix de polices pour comparaison¹¹⁶.

Linéale	Contreformes
abcde abcde <small>TeX Gyre Termes</small> <i>éviter les empattements</i>	aces9 aces9 <small>Arial</small> <i>éviter les contreformes fermées</i>
Interlettrage	Différenciation
nmmnm nmnm <small>Arial</small> <i>éviter les interlettrages trop serrés</i>	Illaog Illaog <small>Mulish-Regular</small> <i>éviter les formes similaires</i>
Proportions	Diacritiques
abcde abcde <small>Arial</small> <i>éviter les proportions trop larges</i>	àéèïjù àéèijù <small>Roboto</small> <i>éviter les accents trop petits</i>
Modulation	Épaisseur
nmmnm nmnm <small>Lato</small> <i>éviter les fortes modulations de tracé</i>	abcde abcde <small>Libertinus Sans</small> <i>éviter les caractères trop fins</i>
Graisse	Italique
aamm aamm <small>Arial</small> <i>éviter les trop faibles contrastes</i>	oaogo oaogo <small>Libertinus Sans</small> <i>éviter les formes cursives</i>

Notons que le dessin d'une fonte adaptée aux malvoyants doit prendre en compte que le corps utilisé varie entre 20 et 24 points. Le corps de lecture est donc près de deux fois supérieur au corps de lecture pour un lecteur clairvoyant. Dans cette *Lettre*, nous testons le rendu dans un cadre classique, et donc pas forcément le plus adapté à la fonte Luciole. La figure 37 page ci-contre donne toutefois un exemple en corps 20.

116. Les polices utilisées dans l'article original ne sont pas indiquées.

FIGURE 37 – Exemple de texte composé en Luciole corps 20.

Le caractère typographique Luciole a été conçu spécifiquement pour les personnes malvoyantes. Ce projet est le résultat de plus de deux années de collaboration entre le Centre Technique Régional pour la Déficience Visuelle et le studio typographies.fr.

Utilisable largement

En plus de sa licence libre et de sa gratuité, une attention particulière a été portée à ce que cette fonte soit complète. Luciole contient plus de 700 glyphes dans chacune de ses variantes (roman, italique, gras et gras italique), et contient quelques symboles grecs et mathématiques pour la notation des sciences jusqu'au niveau baccalauréat¹¹⁷.

Outre les symboles spéciaux et dédiés aux mathématiques, ces 700 signes couvrent les lettres latines classiques d'Europe avec leurs diacritiques et variantes diverses. Fonte OpenType, Luciole propose les *features* désormais classiques¹¹⁸. Toutefois, il n'y a pas de *stylistic set* (*feature ss0n*), c'est-à-dire de variantes stylistiques. Les exemples 69 à 71 page 106 en donnent des illustrations avec LualATEX.

Comparaison avec d'autres fontes

Lors de leurs expérimentations qui ont montré que Luciole est une fonte très adaptée pour les malvoyants (voir [1]), les auteurs de l'étude ont comparé Luciole avec *Arial*, *Verdana*, *Eido*¹¹⁹, *Open Dyslexie* et *Frutiger* dont vous pouvez voir un échantillon en figure 38 page suivante ; on a ajouté quelques autres linéales (de T_EXLive) pour comparaison.

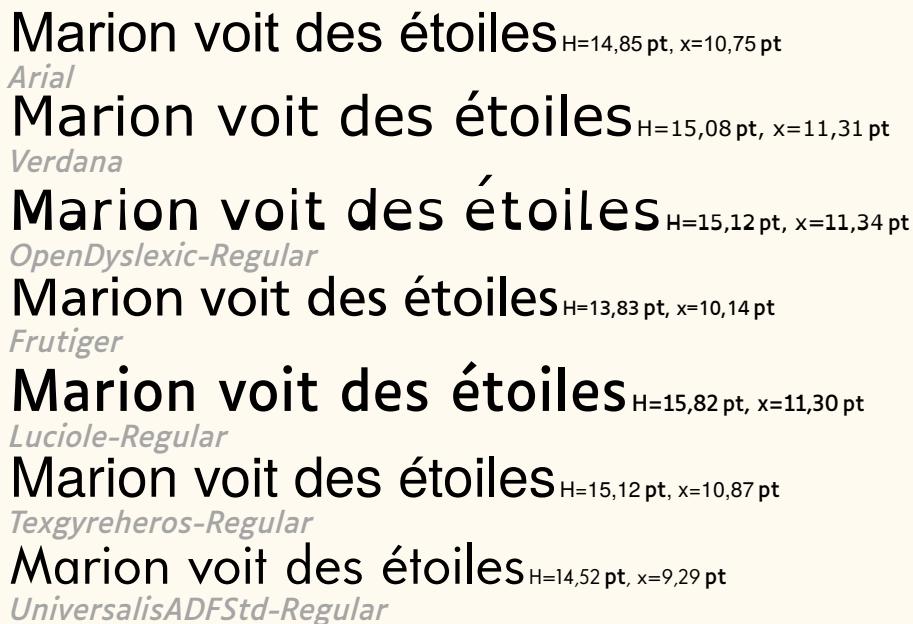
Sans rentrer dans les détails, notons que les tests ont porté sur les préférences subjectives ainsi que sur le temps et la précision de lecture. L'étude visait à voir si Luciole pouvait aider les personnes malvoyantes tout en testant si c'était une fonte aussi adaptée pour tous les lecteurs, malvoyants ou pas. Bien que l'étude demande à être complétée, elle a permis de tester l'effet d'une nouvelle police de caractères conçue spécifiquement pour les personnes malvoyantes sur la lisibilité mesurée objectivement et sur les

117. Comme nous l'avons dit, cette fonte n'est à l'heure actuelle pas *packagée* pour LATEX, mais on peut imaginer qu'elle pourrait même devenir une fonte mathématique en empruntant à une police voisine les caractères manquants. Un sujet d'article et de package pour une prochaine Lettre ? Avis aux amateurs !

118. Voir par exemple celles d'Infini décrites dans la Lettre 45.

119. Malheureusement, nous n'avons pas réussi à obtenir cette fonte.

FIGURE 38 – Illustration des fontes Arial, Verdana, Open Dyslexie, Frutiger et Luciole utilisées dans l'étude [1]; puis d'autres pour comparaison. À droite, valeurs des hauteurs des capitales (H) et d'x (x), en points pour le même corps.



préférences sur papier et à l'écran. Les tests ont aussi été réalisés auprès de participants voyants. La principale conclusion est que Luciole a été préférée aux cinq autres polices pour les personnes malvoyantes (sur papier et écran).

Enfin, les mesures objectives (temps de lecture, précision de lecture et durée de fixation) montrent pour la police Luciole un effet moindre sur la lisibilité. Aucun effet n'a été constaté pour le groupe de personnes à la vision non déficiente.

La figure 38 montre aussi que Luciole est une « grande » fonte (ses capitales sont hautes). Justement, les recherches récentes en matière de lisibilité confirment les toutes premières assertions de Javal, Huey, Legge et d'autres chercheurs selon lesquelles *la taille des caractères* est probablement le plus important facteur physique affectant la lisibilité [3].

Utilisation avec LualATEX

Pour utiliser Luciole avec LATEX, nous avons utilisé LualATEX et le package `fontspec` (on aurait pu utiliser de la même façon XE_LATEX).

Nous avons téléchargé les quatre fichiers depuis le site de ce projet¹²⁰ :

- `Luciole-Regular.ttf`,
- `Luciole-Regular-Italic.ttf`,
- `Luciole-Bold.ttf`
- et `Luciole-Bold-Italic.ttf`.

Ensuite, nous avons configuré la fonte principale et la fonte sans serif avec le code suivant :

120. <https://luciole-vision.com>

Exemple 66

```

1 \usepackage{fontspec}
2 \setmainfont[%]
3   BoldFont={Luciole-Bold.ttf},
4   ItalicFont={Luciole-Regular-Italic.ttf},
5   BoldItalicFont={Luciole-Bold-Italic.ttf},
6   ]{Luciole-Regular.ttf}
7 \setsansfont[%]
8   BoldFont={Luciole-Bold.ttf},
9   ItalicFont={Luciole-Regular-Italic.ttf},
10  BoldItalicFont={Luciole-Bold-Italic.ttf},
11  ]{Luciole-Regular.ttf}
```

Luciole ne fournit pas de fonte à chasse fixe (*mono*). Nous avons choisi de la marier avec *Fira Mono*. Pour cela, nous avons chargé le package *FiraMono* à la suite du code précédent.

Exemple 67

```
1 \usepackage{FiraMono}
```

Enfin, pour permettre la composition de mathématiques, nous avons utilisé le package *firamath-otf* qui configure les mathématiques avec la fonte *Fira Math*, puis nous avons utilisé le package *mathastext* pour remplacer les caractères de texte de *Fira Math* par les caractères de texte de la fonte principale, ici Luciole. Ceci se fait avec le code suivant.

Exemple 68

```

1 \usepackage[fakebold]{firamath-otf}
2 \usepackage[italic]{mathastext}
```

La figure 39 illustre cette configuration. On peut constater que cela ne permet pas d'exploiter complètement la fonte Luciole en mode mathématique. Par exemple, les lettres grecques ne sont pas substituées (mais elles n'existent pas toutes dans la fonte Luciole, c'est donc peut-être plus cohérent...).

FIGURE 39 – Illustration de la composition des mathématiques avec Luciole complétée par la fonte *Fira Math* dont les caractères de texte ont été remplacés par ceux de Luciole.

Theorème 1 (des résidus). Soit f une fonction analytique dans une région G à l'exception des points isolés a_1, a_2, \dots, a_m . Si γ est une courbe dans G fermée simple, rectifiable, positivement orientée qui ne passe par aucun des points a_k , alors

$$\sum_k \text{Res}(f, a_k) = \frac{1}{2\pi i} \int_{\gamma} f(z) dz.$$

Les exemples 69 à 71 page suivante illustrent l'utilisation de *features OpenType* pour accéder à certains caractères de Luciole avec *fontspec*.

Tout d'abord, on a accès aux caractères supérieurs et inférieurs avec l'option

RawFeature=sups ou RawFeature=sinf.

Exemple 69

code

```

1 Lettres supérieures :
2 M{\addfontfeature{RawFeature=sups}me} D.
3
4 Chiffres supérieurs/inférieurs : 12%
5 {\addfontfeature{RawFeature=sups}34}56%
6 {\addfontfeature{RawFeature=sinf}78}90

```

résultat

Lettres supérieures : M^{me} D.
Chiffres supérieurs/inférieurs : 12³⁴56₇₈90

Il est aussi possible d'obtenir le glyphe du zéro barré.

Exemple 70

code

```
Zéro=0, zéro barré={\addfontfeature{RawFeature=zero}0}
```

résultat

Zéro=0, zéro barré=0

Enfin, l'exemple 71 montre que les chiffres peuvent être composés en *old style*, en *lining*, en *proportional* ou *monospaced*.

Exemple 71

code

```

1 \newcommand{\chiffres}{1234567890}
2 \begin{tabular}{|l|l|l|}\hline
3 Chiffres & Elzéviriens & Alignés\\\hline
4 Proportionnels &
5 \addfontfeature{Numbers={OldStyle,Proportional}}
6 \chiffres &
7 \addfontfeature{Numbers={Lining,Proportional}}
8 \chiffres\\
9 Tabulaires &
10 \addfontfeature{Numbers={OldStyle,Monospaced}}
11 \chiffres &
12 \addfontfeature{Numbers={Lining,Monospaced}}
13 \chiffres\\\hline
14 \end{tabular}

```

résultat

Chiffres	Elzéviriens	Alignés
Proportionnels	1234567890	1234567890
Tabulaires	1234567890	1234567890

Luciole est un travail remarquable. C'est une fonte à retenir, et à diffuser largement. Nous espérons qu'elle intéressera quelques membres de notre communauté et qu'ils écriront pour elle le *package* qu'elle mérite.

Jacques André, Patrick Bideault & Maxime Chupin

Références

- [1] Anna Rita GALIANO et al. « Luciole, a new font for people with low vision ». In : *Acta Psychologica* 236 (2023), p. 103926. ISSN : 0001-6918. DOI : <https://doi.org/10.1016/j.actpsy.2023.103926>. URL : <https://www.sciencedirect.com/science/article/pii/S0001691823001026>.
- [2] Jonathan FABREGUETTES et Laurent BOURCELLIER. « Méthode de conception d'un caractère typographique à l'usage des lecteurs déficients visuels ». In : *Revue Graphê* 95 (2023), p. 2-6. URL : <https://www.typographies.fr/N/article-Luciole/article.html>.
- [3] Charles BIGELOW. « Lisibilité et typographie : les recherches durant la première moitié du xx^e siècle ». In : *Histoire de l'écriture typographique – le xx^e siècle*. Sous la dir. de Jacques ANDRÉ. T. I/II. Adverbum, 2016, p. 194-213.



COMPTES RENDUS DE LECTURE

Un passionnant échange



CENTRE NATIONAL DES ARTS PLASTIQUES, éd. *Graphisme en France* 29 (déc. 2023) : *Histoire(s) de design graphique*. ISSN : 2553-629X. URL : <https://www.cna-p.fr/actualites/graphisme-en-france/revues/ndeg29-histoire-s-de-design-graphique-2023>

La nouvelle livraison de la revue annuelle *Graphisme en France*, publiée par le Centre national des arts graphiques et disponible gratuitement, contient pour seul article une longue et passionnante conversation entre les designers graphiques Sara de Bondt¹²¹, Catherine Guiral¹²² et Alice Twemlow¹²³. Cette lumineuse conversation, étayée par la bibliographie de l'article, vous donnera l'envie d'approfondir par d'autres lectures.

On y trouve notamment l'échange suivant, commencé par Catherine Guiral et poursuivi par Alice Twemlow, qui lui répond :

121. Sara de Bondt est designer graphique et chercheuse. Elle enseigne à l'école cantonale d'Art de Lausanne et à l'Académie royale des Beaux-Arts de Gand.

122. Catherine Guiral de Trenqualye est designer graphique et chercheuse. Elle enseigne à l'École nationale supérieure d'Art et de design de Nancy ainsi qu'à la Haute École d'Art et de design de Genève.

123. Alice Twemlow est professeure titulaire de la chaire Wim Crouwel d'histoire, de théorie et de sociologie du design graphique et de la culture visuelle à la faculté des sciences humaines de l'université d'Amsterdam.

« [...] il faut absolument confronter les propos de la personne interviewée aux archives quand un entretien est voué à être diffusé. Une fois publié, ce n'est plus une conversation, mais un document d'archives que les lecteurs utiliseront comme une source. Il y a toujours des modifications et des rectifications à apporter.

– En réalité, la mémoire subjective est aussi légitime. C'est le travail de l'historien, vous ne trouvez pas ? Il faut naviguer entre deux versions de la vérité – la vérité des archives et celles de la mémoire. »

Nous apprécions par ailleurs le fait que *Graphisme en France* soit désormais doublé d'une édition en anglais, ce que nous n'hésitons pas à rapprocher de la tenue, en anglais, d'un événement comme TypeParis¹²⁴. Le monde du design graphique français est ouvert à l'international et c'est tant mieux.

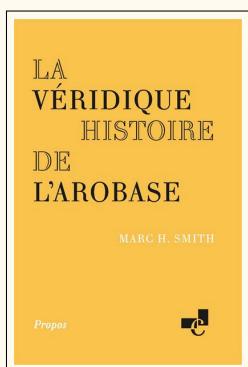
Enfin, ce dernier numéro est composé¹²⁵ en Muoto et Muoto Mono, deux fontes de Matthieu Cortat éditées par 205TF¹²⁶, une dynamique fonderie de Lyon dont le travail est furieusement intéressant.

Adoncques nous recommandons-vous vivement cette dernière livraison.

<https://www.cnap.fr/actualites/graphisme-en-france>

Patrick Bideault

La véridique histoire de l'arobase



Marc SMITH. *La véridique histoire de l'arobase*. Paris : Éditions de l'École nationale des chartes, août 2023. 96 p. ISBN : 9782357231795. 9 €

À la fin des années 1980, un sondage sur la liste GUT avait montré que pour beaucoup des premiers membres de GUTenberg, le signe @ était un caractère spécial d'[ASCII](#), peu utilisé – sauf, justement car peu utilisé, dans des langages informatiques comme Algol-68 (1968) ou Pascal (1970) et bien sûr dans [TeX](#) –, dont on ignorait le nom et l'origine, qu'on avait toujours du mal à trouver sur les claviers, et que l'on ne savait même pas bien dessiner à la main. Puis, Internet oblige, il nous est devenu familier et on a vu écrites plein de choses sur cette « arrobe », parfois exactes, beaucoup plus souvent erronées, farfelues, voire délirantes...

Marc Smith, professeur de paléographie et histoire de l'écriture à l'École des chartes et à l'École pratique des hautes études, a fait des recherches d'historien sur les origines de ce signe dont il présenta les résultats dès 2013 dans une vidéo¹²⁷. Dix ans plus tard, il ressort cette étude, revue et

124. TypeParis est un programme d'enseignement doublé d'un ensemble de conférences, le tout consacré à la création de caractères. Des créateurs originaires de toutes les parties du monde y interviennent, et c'est une occasion unique de les entendre à Paris. <https://typeparis.com/>

125. Le fait que notre modeste publication et *Graphisme en France* changent de fonte à chaque livraison crée une petite similitude entre nos ouvrages ; elle ne nous déplaît aucunement.

126. <https://www.205.tf/>

127. Marc H. Smith, conférence « La véridique histoire de l'arobase » du 29 janvier 2013, Paris, École nationale des chartes, 2013, durée 1h09, [YouTube](#).

augmentée, cette fois dans un opuscule papier¹²⁸.

Marc Smith a commencé par étudier ce signe sous tous ses aspects et relève, dans les premières pages, un peu tout ce qu'on a pu raconter sur cet @, son nom, sa forme, ses ressemblances et son usage dans le monde présent et passé.

De là, il a privilégié deux voies de recherche. L'une, qu'il appelle la piste franco-ibérique, le mène, en remontant le temps, du mot arrobe des imprimeurs français à la typographie espagnole et de là aux manuscrits espagnols, avec une « abréviation à tout faire » (cursive du XIV^e siècle), puis à une préposition « à » que l'on va retrouver en Italie (par exemple chez Gresci au XVI^e siècle) et qui va se répandre en Europe puis en Amérique. La seconde piste, dite anglo-américaine, est balisée par l'emploi de la touche @ de nos claviers et, plus tôt, des machines à écrire, remontant aux livres de comptabilité américains du début des années 1800 ou aux abréviations des traités calligraphiés anglais du milieu du XVIII^e siècle. De ces histoires a priori divergentes, Marc Smith arrive à tirer une synthèse qu'il termine par un *Envoi* d'à peine 5 pages que je me garderai bien de résumer tellement il est dense.

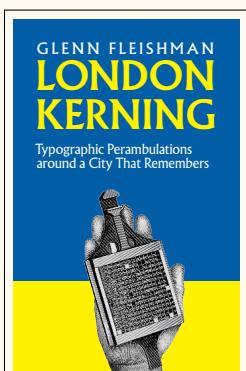
On ne sait qu'admirer le plus de la richesse des documents analysés ou de la démarche (remonter dans le temps les diverses pistes, éliminer ce qui n'est pas sérieux, et terminer par un « envoi » synthétique). Bien sûr, il y aura sûrement plein de gens qui trouveront quelque détail à ajouter, mais a priori ce seront autant de coups d'épée dans l'eau...

Jacques André



EN BREF

Crénage à Londres



Glenn FLEISHMAN. *London Kerning*. Seattle, Washington, États-Unis d'Amérique : Aperiodical llc, 2018. 80 p. ISBN : 978-0-9994897-3-4

L'auteur Glenn Fleishman a mis son ouvrage *London Kerning* en libre accès sur internet. Il s'agit d'un portrait typographique de Londres. Il commence par l'identité graphique de la ville elle-même, qui doit tant à Berthold Wolpe, auquel le livre est dédié. C'est un vrai plaisir que de retrouver son élégant caractère Albertus dans les illustrations de l'ouvrage.

L'auteur présente également la bibliothèque Saint Bride, dédiée aux travaux d'imprimerie, qui contient notamment des originaux d'Eric Gill, et la *Type Archive*, la collection nationale de fonderie de caractères, qui comprend notamment les archives de Monotype.

128. Voir <https://www.chartes.psl.eu/editions/catalogue-des-publications/la-veridique-histoire-de-larobase>.

Marc Smith a lui-même commenté son livre dans diverses interviews dont <https://www.radiofrance.fr/franceinter/podcasts/le-code-a-change/le-code-a-change-3-5971499> et <https://www.rfi.fr/fr/podcasts/de-vive-s-voix/20231025-d-u-moyen-âge-à-nos-jours-la-surprenante-histoire-de-l-arobase>.

Le livre recèle d'autres trésors, et sa bibliographie est précieuse. Si vous lisez l'anglais, n'hésitez pas ! Même un non-anglophone appréciera les nombreuses illustrations !

<https://glog.glennf.com/london-kerning>

Patrick Bideault

GUTenberg à la Journée du Libre Éducatif



Vendredi 29 mars 2024 s'est tenue à la Faculté de droit de l'Université Paris-Est Créteil la troisième édition de la journée du Libre éducatif¹²⁹.

Comme l'indique son site internet,

la Journée du Libre Éducatif a pour objectif d'accueillir aux communs numériques et de soutenir l'écosystème de celles et ceux qui créent et partagent des logiciels et ressources éducatives libres utiles à la communauté scolaire.

Ces journées sont organisées sous la direction d'Alexis Kauffmann qui est un libriste bien connu. Ancien professeur de mathématiques, il est à l'origine de Framasoft en 2001. En 2021, il rejoint, au ministère de l'éducation nationale, la direction du numérique pour l'éducation¹³⁰ en tant que chef de projet logiciels et ressources éducatives libres.

Cette édition s'est déroulée en deux parties¹³¹ :

- la matinée en séance plénière, très dynamique, avec un exposé inaugural et de nombreuses présentations *flash* de cinq minutes par projet libre, tous très intéressants.
- l'après-midi, organisée en de nombreuses sessions parallèles, avec des exposés de trente minutes autour de thématiques très variées.

Le public a été principalement constitué d'enseignants et d'enseignantes, de l'école à l'université, mais tout de même principalement du collège et du lycée.

C'est pendant les exposés de l'après midi que j'ai représenté l'association aux côtés de Cédric Pierquet et Christophe Poulain, autour de l'atelier : « LATEX ? Je m'y mets dès demain ! ». Nous sommes très heureux que LATEX ait été représenté dans ce contexte par ces deux contributeurs d'exception.

129. <https://journée-du-libre-éducatif.forge.aeif.fr>

130. https://fr.wikipedia.org/wiki/Direction_du_num%C3%A9rique_pour_l'%C3%A9ducation

131. Le programme est consultable ici à l'adresse suivante : <https://journée-du-libre-éducatif.forge.aeif.fr/assets/pdf/JDLE2024-programme.pdf>.

L'expérience a été intéressante et ce rendez-vous, désormais annuel, est une très bonne chose pour le logiciel libre et nous ne pouvons que soutenir un tel événement.

Vous pouvez consulter sur la page dédiée de notre site web le diaporama que nous avons présenté :

https:
[//www.gutenberg-asso.fr/Journee-du-Libre-Educatif-2024](https://www.gutenberg-asso.fr/Journee-du-Libre-Educatif-2024)

Maxime Chupin



⌚ RÉBUS

Nostra culpa maxima : dans la Lettre 51, nous avons oublié de publier la solution du rébus publié dans la Lettre 50 !

La voici.

Pour rappel, ce rébus était rédigé comme suit :

☞ Rébus (*un toast*)

Mademoiselle, à vore !

Solution ci-dessous

☞ Solution du rébus de la Lettre 50

Mademoiselle Sophie (sauf i), à votre santé (sans t).

Yvon Henel



⌚ PROCHAINES RENCONTRES

On nous a reproché de ne pas publier de calendrier des prochains événements, et ce reproche était fondé. Voici donc les prochaines rencontres concernant nos logiciels préférés :

- la conférence BachoTeX 2024 aura lieu à Bachotek (Pologne) du 1^{er} au 5 mai prochains. Son thème est, en anglais, *composed thoughts*, ce que l'on pourrait traduire par *pensées composées*, mais y lirait-on la référence au travail de composition typographique ? *Pensées de composition* n'est pas plus heureux. Les traducteurs en herbe, de préférence polonophones et n'ignorant rien de l'heuristique, qui désireraient nous aider sur ce point sont invités à se manifester auprès du secrétariat. Et ceux d'entre vous qui désireraient se rendre

sur les rives bucoliques du lac Bachotek trouveront les informations nécessaires sur la page dédiée : <https://www.gust.org.pl/bachotex/2024-en>

- le TUG 2024 aura lieu à Prague (République tchèque), fin juillet 2024. La page <https://tug.org/tug2024/> sera mise à jour petit à petit et reprendra toutes les informations utiles.
- les utilisateurs de ConTeXt sont invités à se rendre à Lutten/Hardenberg (Pays-Bas) du 17 au 23 août 2024, pour de studieuses journées de travail. Plus d'informations ici : <https://meeting.contextgarden.net/>
- les grapholinguistes du monde entier se retrouveront à Venise du 23 au 25 octobre, et leur conférence est par l'ACL¹³² et l'Association typographique internationale¹³³. Donald Knuth y prendra la parole. <https://grafematik2024.sciencesconf.org>
- et la prochaine journée GUTenberg aura lieu durant l'automne. D'ici là, retrouvons-nous en ligne pour nos exposés mensuels !



ACRONYMES

AG Assemblée Générale

ASCII *American Standard Code for Information Interchange* (code américain normalisé pour l'échange d'information)

BBB *BigBlueButton*

CA Conseil d'Administration

CI/CD *Continuous Integration/Continuous Delivery* (intégration continue/livraison continue)

CSS *Cascading Style Sheets* (feuilles de style en cascade)

CTAN *Comprehensive TeX Archive Network* (réseau complet d'archives TeX)

CV Curriculum Vitæ

DOI *Digital Object Identifier* (identifiant numérique d'objet)

EDI Environnement de développement intégré

ENS École Normale Supérieure

EPUB *electronic publication* (publication électronique)

FAQ *Frequently Asked Questions* (questions fréquemment posées, souvent librement traduit en « foire aux questions »)

FUBAR *Fucked Up Beyond All Recognition* (état psychique habituel d'un utilisateur de système WYSIWYG de composition de documents)

HTML *HyperText Markup Language* (langage de balises pour l'hypertexte)

ISBN *International Standard Book Number* (numéro international normalisé du livre)

ISSN *International Standard Serial Number* (numéro international normalisé des publications en série)

132. ACL pour *Association for Computational Linguistics*, en français « Association pour la linguistique informatique » – https://fr.wikipedia.org/wiki/Association_for_Computational_Linguistics

133. <https://atypi.org/>

NdT Note du traducteur

OMS Organisation mondiale de la santé

OTF *Open Type Format* (format Open Type)

PDF *Portable Document Format* (format de document portable)

PNG *Portable Network Graphics* (graphiques portables pour le réseau)

PV Procès Verbal

REST *ReStructuredText*

TikZ *TikZ ist kein Zeichenprogramm* (TikZ n'est pas un programme de dessin)

TUG *TEX User Group* (groupe international d'utilisateurs de *TEX*)

URL *Uniform Resource Locator* (localisateur uniforme de ressource ou, plus clairement, format des adresses du Web)

WYSIWYG *What You See Is What You Get* (ce que vous voyez est ce que vous obtenez)

XML *eXtensible Markup Language* (langage de balisage extensible)

YAML *YAML Ain't Markup Language* (YAML n'est pas un langage de balisage (acronyme récursif))

La rédaction tient à remercier Damien Debril, Yann Denichou et MFerrer pour les erreurs qu'ils ont remarquées dans la *Lettre 51*; ces erreurs ont été corrigées. Si vous en remarquez d'autres, merci de nous les signaler par courriel à secretariat@gutenberg-asso.fr.

La rédaction rappelle au lecteur que le code source de la présente revue est inclus *dans* le présent **PDF**. On y accède en fin d'article en cliquant sur les trombones.

Par leurs textes, leurs suggestions ou leurs corrections, et par ordre alphabétique de la deuxième lettre de leurs patronymes, ont contribué à cette *Lettre* : Damien Debril, Yvon Henel, Yann Denichou, MFerrer, Céline Chevalier, Maxime Chupin, Patrick Bideault, Cédric Pierquet, Denis Bitouzé, François Druel et Bastien Dumont.

La Lettre GUTenberg

Bulletin irrégulomestriel de l'association GUTenberg

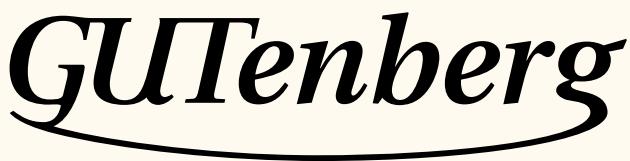
Directeur de la publication : Patrick Bideault

Comité de rédaction : Patrick Bideault, Denis Bitouzé,
Céline Chevalier, Maxime Chupin
& Bastien Dumont

Adresse de la rédaction : Association GUTenberg

15 rue des Halles – BP 74
75001 Paris

ISSN : 2742-6149 (version numérique)



Association GUTenberg
15 rue des Halles – BP 74
75001 Paris
France
secretariat[at]gutenberg-asso[dot]fr

Site Internet : <https://www.gutenberg-asso.fr/>

Cahiers : <http://www.numdam.org/journals/CG/>

Problèmes techniques :

la liste gut : <https://www.gutenberg-asso.fr/-Listes-de-diffusion->

le site TE^Xnique de questions et réponses : <https://texnique.fr/>

la foire aux questions : <https://faq.gutenberg-asso.fr/>

Cette association est la vôtre : faites-nous part de vos idées, de vos envies, de vos préoccupations à l'adresse secretariat[at]gutenberg-asso[dot]fr.

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 2024

Les membres de GUTenberg peuvent adhérer à l'association internationale, le **TUG**, 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**¹³⁴ (IBAN : FR76 1870 7000 3003 0191 3568 475)

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

- **Paypal**¹³⁴ : <https://www.gutenberg-asso.fr/?Adherer-en-ligne>
- **bulletin et chèque**¹³⁴ : <https://www.gutenberg-asso.fr/?Adherer-a-l-association>

134. Nous vous remercions de privilégier le **virement bancaire**.