

Retex

7 décembre 2025

Retex est un format minimaliste et évolutif du langage Tex, dédié à la composition typographique. Il propose une syntaxe non invasive, un support unicode et des performances optimales. Il a pour ambition de simplifier au maximum la saisie des textes à mettre en forme.

Tex et son écosystème

Tex est un langage de macros qui permet la production de documents destinés à l'impression. Conçu par Donald KNUTH et ses étudiants dans les années quatre-vingt, il est documenté dans le volume A « Computers & Typesetting » de son ouvrage *The Texbook*. Il est un standard de fait de l'édition scientifique dans de nombreuses disciplines car il permet tout particulièrement la mise en page soignée des mathématiques. Il est reconnu pour la qualité typographique résultante de ses algorithmes ; il prend en charge la mise en forme automatisée du texte : justification et césure des lignes, placement des paragraphes et références croisées.

Un moteur Tex est un programme qui interprète un fichier texte brut : une suite de caractères qui respecte la syntaxe du langage et ne présente aucune mise en forme à priori. Le moteur exécute des macros, directives fournies en amont ou avec le texte à composer, pour mettre en page ce dernier ; il produit un document graphique destiné à l'impression, bien que les usages aient évolué vers la lecture sur écran. Le moteur historique est décrit dans le volume B « Tex: The Program » ; son successeur Pdftex, ancien mais toujours d'usage, implémente la micro-typographie ainsi que la génération PDF par défaut ; Xetex est un dérivé qui prend en charge l'encodage de fontes modernes, plutôt que le système originel Metafont ; enfin Lualatex y ajoute l'intégration d'un langage de programmation. Tous suivent les algorithmes de composition qui ont fait le succès du premier moteur.

Les formats sont des fichiers pré-chargés et constitués d'une collection de macros et de commandes. Ils ont pour fonction de préparer la composition et pour ambition de faciliter la rédaction. Le format original est le Plain_{tex} ; peuvent être cités Eplain, Amstex, Optex et dans une certaine mesure Context. Le format Latex domine très largement malgré la diversité des alternatives. Mais il n'est pas satisfaisant : ses performances sont médiocres ; sa syntaxe est verbeuse ; les choix esthétiques relèvent trop souvent des concepteurs de “classes”, qui ne laissent aux rédacteurs que très peu de marges de manœuvre.

C'est en retour d'expérience de ce format populaire et donc avec l'ambition d'éliminer ses lacunes que je me suis lancé dans le projet Retex, en premier lieu pour satisfaire un besoin spécifique... Sans rivaliser en fonctionnalités ou disponibilité de la documentation, le format que je propose cible avant tout celles et ceux qui connaissent Latex, en ont perçu les limitations, et souhaitent explorer davantage les arcanes du monde Tex, sans pour autant perdre la faculté de produire des documents simple aussi, voir plus, facilement qu'avec Latex. Aussi un·e débutant·e aventureux·se et particulièrement motivé·e pourra se lancer et avoir l'opportunité d'apprendre Tex progressivement. Le choix du minimalisme est le pivot autour duquel s'articule le format, car il est le garant d'une pratique de l'informatique transparente, évolutive et autonome.

Retex distingue et sépare très strictement la rédaction du document, dans le fichier source, et la composition proprement dite, ou style. La syntaxe de rédaction se veut la plus discrète possible : elle repose sur un langage qui n'interfère pas avec le texte de l'auteur. Pour autant, le style se décrit avec toute la puissance et la flexibilité du langage de macros Tex, sans abstraction et dans un dialecte proche de Plain.

La vocation de Retex est de proposer une syntaxe du source dans la lignée des langages de balisage léger tels que Markdown, Restructuredtext, etc. De plus, le format s'appuie sur la saisie Unicode avec un encodage UTF8, y compris avec le moteur historique, pour restituer de manière transparente la typographie du fichier source. Le rédacteur peut donc investir

une méthode de saisie moderne : Bépo, Ibus, etc. pour exploiter pleinement le potentiel de Retex. Le choix d'un format fidèle permet à l'utilisateur une grande liberté typographique ; la contrepartie est qu'il en devient pleinement responsable et se doit d'y être sensibilisé.

Le format est adapté au français : les espaces sont toujours simples, l'indentation directement saisie par le rédacteur à sa convenance. L'alphabet et les glyphes spécifiques à la langue sont pris en charge. Les césures suivent les règles propres au français. Retex s'inspire, avec quelques libertés, de l'extension bien connue Babel de Latex.

Le style par défaut offre un exemple simple, qui propose donc une mise en page que l'utilisateur peut adapter en élaborant son propre style. Le format propose très peu de macros ; au contraire il encourage l'apprentissage de Tex et de ses primitives pour créer les styles souhaités. Le format autorise ainsi l'utilisateur à acquérir une véritable expertise de la composition et l'incite de ce fait à l'autonomie. De plus, il apporte la possibilité d'automatiser les tâches répétitives — Tex est un langage Turing-complet.

Enfin, le format se doit d'être totalement compatible avec le moteur original, pour les meilleures performances, et Pdftex à minima. L'enjeu est d'obtenir le flux de travail le plus fluide possible sans sacrifier la qualité de la composition finale. Le style doit pouvoir être intégré dans le format, par chaînage, pour former une extension. Enfin le travail "par partie" sur des projets complexes, avec de nombreux fichiers source, doit être possible : ainsi, le rédacteur peut par exemple se concentrer sur un chapitre, sans avoir à composer tout son livre pour apprécier le résultat.

Utiliser Retex

Un exemple minimal de document contient : un préambule, une ligne vide, le texte de labeur et se termine *impérativement* par une ligne vide suivie de l'esperluette &. Le préambule se lit ligne par ligne : un titre et des sous-titres. Le texte de labeur se scinde en paragraphes qui commencent après toute ligne vide. Cela seul, et rien que cela, suffit à rédiger un document Retex valide ; ainsi nul besoin d'apprendre une syntaxe complexe. Pour des documents plus élaborés : les alinéas sont matérialisés par des tabulations ; un début de paragraphe signalé par la barre verticale | positionnera le reste de la ligne en titraille. Enfin le rédacteur a la possibilité d'écrire des commentaires, exclus du document final.

Plus généralement, le format exploite quelques caractères de la table ASCII, d'usage douteux en typographie, pour encoder des directives de mise en forme. Le format s'inspire aussi de Troff — un logiciel plus ancien de composition — en prenant avantage de ce que la ponctuation ne peut débiter un paragraphe. Le tableau ci-après synthétise la syntaxe : selon que le caractère apparaît après une ligne vide — colonne "Paragraphe" —, ou directement dans le "Texte" de labeur, il prendra la signification de l'entrée correspondante. La colonne "Borne" indique quand la mise en forme prend fin ; ¶ signifie qu'elle reste valable sur tout le paragraphe : prochaine ligne vide ou alinéa selon le contexte. Enfin la dernière colonne sera utile au développement de nouveaux styles.

La composition du document nécessite tout d'abord de préparer les fontes, qui peuvent éventuellement être installées dans le répertoire utilisateur ou système texmf. L'exemple donné ci-après postule que ce n'est pas le cas, et par conséquent elles sont construites dans le répertoire /tmp via le script make fourni avec les sources. Ensuite il faut écrire le dump du format et de ses extensions, grâce à l'option -ini et en chaînant les étapes si nécessaire. Enfin le document est composé en faisant appel au format voulu avec l'option -fmt.

```
1 Titre
2 Auteur
3
4 | Section
5 Un paragraphe.
6
7
8 &
```

```
# Fontes virtuelles pour le français (depuis les sources et vers /tmp)
: ./make font && export TFMFONTS=:/tmp
# Produire le format Retex (./make retex)
: tex --output-directory /tmp -ini retex.ini
# Chaînage d'extension(s) (disponibles: ./make math; ./make verbatim)
: tex --output-directory /tmp \
    -ini -jobname retex(-ext1)-ext2 \&retex(-ext1) ext2.ini
# Composer un document (avec ou sans extension)
: tex --output-directory /tmp -fmt retex(-extension) document.tex
```

	Paragraphe	Texte	Borne-s	Signature-s (à faire)
HT	Alinéa	Alinéa		<code>\tab</code>
"	?	Guillemets		<code>\quot</code> (groupe)
#	Commentaire	Commentaire	Ligne	
&	Fin de source	Caractère		<code>\end</code>
'	?	Apostrophe		
*	Index ?	Énumération	Alinéa	
+	Table	Caractère	¶	<code>\align</code>
-	?	Liste	Alinéa	Alt. — ?
/	Citation	Emphase	¶ /	<code>\sl \em</code>
@	Note ?	Référence ?	Mot	
\	Listing	Verbatim	¶ \	<code>\tt</code> (ext. verbatim)
^	?	Exposants	Mot	
˘	?	Mathématiques	¶ ' extension math	
	Section	Petites capitales	Ligne Mot	<code>\section#1^M \sc</code>

Entre autres, les caractères Unicode reconnus sont : les caractères accentués français, les guillemets à la française et à l'anglaise, apostrophe, points de suspension, espaces insécables justifiantes fine et normale, espace quadratin, suggestion de césure, tirets (demi-)quadratin. La liste se complète selon mon propre besoin et la disponibilité dans l'encodage sélectionné.

Un ¶ à rédiger sur l'environnement, indispensable à la fluidité du travail.

1. Unix : Xdefaults pour xdvi, Bépo (normalisé Afnor), Xcompose.
2. Sur le *hack* et la chaîne d'encodage : clavier, UTF8, *texnansi* et *virtual font*. Nécessaire sur la gestion des *virtual fonts* pour la génération des documents. À intégrer dans le répertoire de travail `--output-directory`, et `dvicopy` pour un fichier portable.
3. Texlive (installation minimale pour la performance).
4. Vim (compilation par pipe, let engine, coloration syntaxique, non affichables).
5. Tex (fontes entre autre? `fc-list` ou fonte Tex, intérêt d'une installation minimale).

Développer

Retex n'est pas abouti ; si la syntaxe et le style sont stabilisés, tout n'est pas implémenté, y compris des fonctionnalités essentielles et attendues. Dans tous les cas, pour travailler avec ce format il est impératif de conserver la version exploitée et ses extensions avec votre projet, car contrairement à la tradition (La)Tex le format se conçoit comme extensible. Reste à faire, entre autres :

- continuer à commenter & organiser le code du format, séparation syntaxe–style ;
- extension mathématique en cours, comme un ``groupe'`, *display* avec tabulation ;
- références `@class:key`, avec `class : toc (auto.)`, `idx`, `bib`, ... et notes de pied de page `@` suivi d'une ponctuation en texte, puis `@` en ¶ (si compatible `\output`) ;
- documents multi-colonnes ;
- césure de page & position de la titraille, satisfaisante ? ;

- alignement optique exhaustif, travail à faire sur les PDF et le moteur Pdftex, et support de Luatex à minima ;
- *virtual font* : césures et crénage des cas ‘œ’, ‘Œ’ et ‘ÿ’ ;
- notion de famille de fonte, `\lang[0-15]`, plus simple et plus souple que NFSS car n’exige aucun postulat ;
- un script d’appel à `dviselect|dviconcat`, pour réordonnancer les pages — nécessite de normer les pages dvi, compteurs 0 à 9 ;
- saisie multi-fichiers, avec niveaux de titrilles en cascade ;
- verbatim comme exemple de langage ?, absence de césure & famille de fontes ;
- délimitation par espaces insécables pour l’exposant (par exemple M^{me}) ;
- création de compteurs à la volée, anonymes ou étiquetés, avec possibilité de forcer la valeur, au format `*[0-9]...[a-z]...`, pour conserver la syntaxe des énumérations ;
- peu de tests donc beaucoup de “bugs”, parfois par compromis perf. & simplicité...

Le traitement des références @ sera unifié, mais chaque classe aura son propre style. Elles seront traditionnelles : les tables, y compris le sommaire, seront placées en fin. Ainsi la génération en une seule passe Tex est envisageable ; mais la réorganisation des pages en post-production sera toujours possibles, pour placer une table des matières en préambule.

Une famille de fonte se conçoit comme une liste conventionnelle de bascules de fonte, par exemple : `\rm`, `\it`, `\sc`, &c. pour du texte de labeur ; une sélection de famille redéfinit cette liste, avec un registre d’allocation mis à disposition. Par exemple `\langX` créera l’alias `\rm -> \rmX`. Une fonte est intimement liée à un alphabet et donc tout ce qui en dépend : langue donc césures via primitive `\language` et (trans)codages entrée-sortie, le tout étant, là encore, unifié sous la même famille commandes.

Un ¶ sur les extensions. C’est un fichier qui débute par le code ASCII `esc '33`, pour réactiver le caractère d’échappement Tex `\`. Deux extensions sont livrées : verbatim & math.

Le concepteur de Retex peut être contacté par courriel à l’adresse `nicolas@gaudin.ovh`.