

Enrichir les capacités de Quarto avec les filtres Lua

Christophe Dervieux¹

Résumé (max 300 mots)

Dans le cadre de la production de rapports institutionnels ou de publications scientifiques, les utilisateur·rice·s de Quarto [1] se heurtent parfois à des besoins spécifiques que les fonctionnalités natives ne couvrent pas. Qu'il s'agisse d'automatiser des règles de mise en forme complexes ou de modifier la structure profonde d'un document, les filtres Lua constituent la solution de choix pour adapter les fonctionnalités de Quarto à ses besoins.

Cette présentation vise à démystifier l'utilisation de Lua au sein de Quarto. Bien que les non-développeur·euse·s les considèrent souvent hors d'atteinte, les filtres Lua [2] reposent pourtant sur une logique claire : la manipulation de l'Arbre de Syntaxe Abstraite (AST) [3]. Nous suivrons une progression pédagogique claire : du script externe au filtre Lua intégré, puis à l'extension Quarto réutilisable. Nous explorerons comment Quarto « voit » un document comme une suite de blocs et d'éléments en ligne, et comment de petites fonctions ciblées permettent de transformer ces éléments de manière programmatique.

À travers des exemples concrets — de la simple substitution de texte à la modification d'en-têtes — nous montrerons comment cette mécanique résout le problème du « je suis bloqué ». Aucune expertise en programmation n'est requise : quelques lignes de code suffisent pour automatiser des tâches répétitives. L'objectif est de donner aux participant·e·s les bases nécessaires pour créer leurs propres extensions et automatiser leurs flux de travail, rendant ainsi Lua accessible à toute la communauté Quarto.

Mots-clefs (3 à 5) : Quarto - Développement - Reproductibilité - Lua - Filtres - Automatisation

Développement

Nous structurerons notre présentation de 20 minutes autour des points clés suivants.

La situation de blocage

Le point de départ est une tâche manuelle répétitive que l'on ne peut pas automatiser. Par exemple : ajouter systématiquement un symbole TM après chaque mention d'une marque, formater des abréviations, ou appliquer des règles typographiques spécifiques. Nous montrerons comment passer d'un script de post-traitement externe à une logique intégrée au moteur de rendu pour transformer le document pendant la compilation sans modifier le fichier source.

Comprendre l'AST (Abstract Syntax Tree)

Nous rendrons accessible l'aspect technique central. Nous montrerons que Quarto ne manipule pas du texte brut, mais une structure d'objets organisée en blocs (paragraphes, en-têtes, blocs de

code) et éléments en ligne (texte, liens, emphase). Une fois compris que le document se compose d'éléments de types `Str`, `Header`, ou `CodeBlock`, l'écriture d'un filtre devient une simple règle de correspondance : « pour chaque élément de type X, appliquer la transformation Y ».

Démonstration de la simplicité de Lua

Contrairement aux idées reçues, Lua est un langage léger et lisible. Les filtres Lua sont, au fond, de simples fonctions qui parcourent le document et peuvent modifier, remplacer ou supprimer des éléments, voire déclencher d'autres actions. Nous présenterons un exemple minimaliste de filtre capable d'identifier un motif textuel et de le transformer (par exemple, remplacer automatiquement un terme par sa version avec symbole spécial). Une dizaine de lignes de code suffisent pour résoudre un problème qui nécessiterait autrement une intervention manuelle répétée à chaque compilation.

Vers la réutilisation : Les extensions Quarto

Enfin, nous aborderons la progression naturelle vers la réutilisation : transformer un filtre ad-hoc en extension Quarto [4]. Cet empacotage permet de distribuer des solutions prêtes à l'emploi dans une équipe ou une communauté, avec une simple ligne dans le YAML du document. Cela renforce la reproductibilité et l'homogénéité des publications, tout en masquant la complexité technique aux utilisateur·rice·s finaux. Nous soulignerons également l'existence d'un écosystème communautaire riche qui offre déjà de nombreuses extensions pouvant servir d'inspiration ou de solutions directement réutilisables.

Conclusion et perspectives

En conclusion, cette présentation vise à encourager les participant·e·s à considérer les filtres Lua comme une solution accessible la prochaine fois qu'ils-elles se sentiront bloqué·e·s face à une limitation de Quarto. Nous invitons donc les participant·e·s à essayer. Avec une compréhension de base de l'AST et quelques lignes de code, il devient possible d'automatiser des tâches complexes et de personnaliser profondément les documents. Les filtres Lua ne sont pas réservés aux expert·e·s : ils constituent un outil pratique et puissant à la portée de toute la communauté Quarto.

Références

Bibliographie

- [1] J. Allaire, C. Teague, C. Scheidegger, Y. Xie, C. Dervieux, et G. Woodhull, « Quarto ». [En ligne]. Disponible sur: <https://github.com/quarto-dev/quarto-cli>
- [2] Pandoc, « Pandoc Lua Filters ». [En ligne]. Disponible sur: <https://pandoc.org/luafilters.html>
- [3] J. MacFarlane, A. Krewinkel, et J. Rosenthal, « Pandoc ». [En ligne]. Disponible sur: <https://github.com/jgm/pandoc>
- [4] Quarto, « Filters ». [En ligne]. Disponible sur: <https://quarto.org/docs/extensions/filters.html>