
N° 3 | 2025
2025

Programming Historian en français : les défis de la fabrication d'une revue francophone sur les humanités numériques

Alexandre WAUTHIER *Ingénieur d'études*
Service Accompagnement des projets et science ouverte
UAR 2011
Campus Condorcet
Daphné MATHELIER

Marie FLESCH

Édition électronique :

URL :

<https://demc-journal.org/articles/revue-3/3351-programming-historian-en-francais-les-defis-de-la-fabrication-dune-revue-francophone-sur-les-humanites-numeriques>

ISSN : 3036-5295

Date de publication : 19/12/2025

Cette publication est sous licence **CC BY-NC-ND** (Attribution - No commercial - No derivatives).

Pour **citer cette publication** : WAUTHIER, A., MATHELIER, D., FLESCH, M. (2025) Programming Historian en français : les défis de la fabrication d'une revue francophone sur les humanités numériques. *DEMC Journal*, (3). <https://doi.org/10.34745/>

Mots-clés :

Introduction

En 1971, Edward Shorter, historien de la médecine à l'Université de Toronto publia *The Historian and the Computer: A Practical Guide*. Dans l'introduction de son ouvrage, il constate que « rien n'existe actuellement pour l'historien qui veut utiliser un ordinateur, mais n'a pas la moindre idée de comment procéder¹ ». Le livre guidait le lecteur pas à pas à travers le processus de conception, de mise en œuvre et d'interprétation des résultats d'une étude historique quantitative exploitant les ordinateurs centraux IBM de l'époque. Selon Adam Crymble (2021), Shorter démystifiait ce que les ordinateurs étaient et pouvaient faire, et fut le premier à replacer tout cela dans le contexte de la profession historique. Il s'agit d'un exemple précoce de documentation à portée autoformatrice d'usages de l'informatique à des fins de recherche en sciences humaines et sociales.

Comme Shorter, d'autres historien·nes ont saisi l'opportunité d'utiliser un ordinateur à des fins de recherche. Dans cette lignée, l'existence même de la revue *Programming Historian* a été rendue possible grâce à la volonté des communautés de pratiques de mettre en commun de ressources éducatives libres (Gille Levenson, Patat, 2022). Ce besoin naissant d'avoir une entraide communautaire a également pris forme dans le monde académique. Ainsi, la recherche et l'enseignement en humanités numériques se situent à l'intersection des technologies numériques et des sciences humaines. Bien que les humanités numériques soient difficiles à définir, on trouve cette notion d'intersection dans le monde anglophone (Terras, 2011) et francophone (Dacos & Mounier, 2015). De manière plus précise, les humanités numériques étudient l'impact de ces technologies sur le patrimoine culturel, les institutions de mémoire, les bibliothèques, les archives et la culture numérique. De plus, les humanités numériques ne se limitent pas à l'usage d'outils informatiques génériques, mais constituent un paradigme à part entière, dans lequel un dialogue entre elles et l'informatique doit être constamment nourri (Longhi, 2019). Ce dialogue s'opère par le développement de méthodes et d'outils spécifiquement pensés pour les besoins et les regards des humanités sur le numérique.

Les humanités numériques ont été sujettes à débat pour en définir les contours et appréhender un changement de paradigme dans le monde de la recherche, à travers l'émergence d'une dématérialisation des habitudes de travail. Initiés aux États-Unis par des étudiants en histoire dès la fin des années 2000, les THATCamp (*The Humanities and Technology Camp*) ont profondément transformé les pratiques des humanités

numériques. En se diffusant rapidement en Europe, notamment en 2010, à Paris, ces non-conférences ont permis d'abattre les barrières institutionnelles et disciplinaires, tout en encourageant l'innovation et la co-construction du savoir (Clavert, 2010). En définitive, la singularité la plus manifeste d'un THATCamp tient à l'insaisissable : elle développe un état d'esprit, la rendant plus vivante et plus collective (Thély *et al.*, 2014). Le caractère informel, léger, collaboratif, peu coûteux, à but non lucratif, spontané, opportun et disponible en ligne a fait la force des THATCamp (McGrath, 2020). Dans différents pays occidentaux, ces derniers ont favorisé la création de réseaux interdisciplinaires ouverts, la circulation des savoirs et l'expérimentation collective de nouveaux outils et méthodes.

Néanmoins, dans le domaine des humanités numériques, largement dominé par l'anglais, le multilinguisme est crucial pour la recherche et l'enseignement : il permet une plus large diffusion des savoirs et méthodes tout en promouvant la diversité des communautés de recherche (Fiormonte, 2021). C'est ce constat qui a poussé la revue *Programming Historian* à lancer trois branches supplémentaires en espagnol, français et portugais (Isasi *et al.*, 2023). *Programming Historian* publie des présentations didactiques et évaluées par les pairs, appelées « leçons ». Elles permettent de s'initier et d'apprendre à utiliser un panel étendu d'outils et de méthodes computationnelles. L'objectif principal est de faciliter la recherche et l'enseignement en sciences humaines et sociales.

Comme ses sœurs hispanophone et lusophone, *Programming Historian en français* publie majoritairement des traductions de leçons publiées en anglais dans *Programming Historian*, et a une identité double : construite sur un projet éditorial anglo-saxon, qui a produit 117 leçons depuis 2008, elle s'adresse à un public francophone tout en prenant en compte les attentes et pratiques de son lectorat dans ses choix d'édition et de publication. Cette contribution propose d'explorer les défis auxquels est confrontée la revue, et en particulier sa branche francophone, qui semble fédérer plus timidement la communauté des sciences humaines et sociales que *Programming Historian en español* et *Programming Historian em português*, avec 35 leçons publiées en français depuis 2019 contre 66 leçons en espagnol depuis 2018 et 54 leçons en portugais depuis 2021 (en juillet 2025).

Nous commencerons par présenter l'approche singulière de *Programming Historian*. Bâti sur les valeurs de la science ouverte et de l'*open source*, le processus éditorial de la revue comprend, entre autres, la rédaction des articles en Markdown par les auteur·es ou traducteur·ices, une évaluation ouverte sur GitHub, une publication sur un site statique et la transparence sur toutes les personnes impliquées dans ce processus. Comment ces nouvelles pratiques s'intègrent-elles dans le contexte de la science ouverte et dans le paysage éditorial francophone des sciences humaines et sociales, à la fois très diversifié et très normé ? Alors que le Web regorge de tutoriels sur les méthodes et outils numériques, comment *Programming Historian en français* fait-elle pour donner une légitimité académique à ses publications, qui sont des « leçons » évaluées par les pairs et non des articles scientifiques « classiques » ? Comment promeut-elle l'inclusivité et la diversité culturelle dans le domaine des humanités

numériques (une des missions de la revue), et quels sont les défis et difficultés rencontrés pour publier et atteindre l'ensemble des communautés francophones ?

Pour étayer ces réflexions, notre contribution s'appuiera, en premier lieu, sur une analyse du processus éditorial de la revue pour publier ses leçons. Nous proposerons ensuite de mettre en perspective les enjeux de la traduction des leçons en matière de coexistence et de pérennité au sein de la revue. Enfin, une dernière partie évoquera les enjeux majeurs de la revue : ceux de la diversité et de l'inclusivité, sous le prisme de l'accessibilité, tant des leçons que des outils numériques.

I) Un processus éditorial singulier dans le paysage de la publication scientifique

Présentation de la revue

Programming Historian est une revue en libre accès, évaluée par les pairs, qui publie des articles, appelés « leçons », pour initier et former aux outils, techniques et méthodes numériques, principalement à destination des chercheurs, enseignants et étudiants en sciences humaines et sociales. Le projet vise à faciliter l'apprentissage des humanités numériques et à rendre accessible la maîtrise de nouveaux outils numériques pour la recherche et l'enseignement. En tant que revue, *Programming Historian* cherche à harmoniser ses flux de publication afin d'assurer la cohérence et la qualité des leçons publiées dans son catalogue multilingue (Quiroga et al., 2024).

Les leçons publiées sur *Programming Historian* constituent un objet singulier, faisant la particularité de la revue. De fait, les leçons vont au-delà du simple tutoriel. Elles s'appuient sur des données authentiques, sont étayées par des réflexions méthodologiques et sont évaluées par les pairs, au même titre que les articles scientifiques classiques. En juillet 2025, la revue comptait 272 leçons — dont 35 pour la branche française — organisées en fonction de thématiques (Python, manipulation des données, gestion des données, etc.) et des cinq principales actions du processus de recherche : acquérir, transformer, analyser, présenter, préserver.

Nous proposons dans cette partie de présenter l'histoire du *Programming Historian* afin d'évoquer ses prémices ainsi que le contexte nord-américain de sa création. Nous nous attarderons ensuite sur les défis relevés par sa branche francophone lors de son lancement en 2019.

Historique

L'histoire de *Programming Historian* commence en 2008 lorsque William J. Turkel et Alan

MacEachern, professeurs au département d'Histoire de l'Université Western Ontario au Canada, publient une première ressource intitulée *The Programming Historian* sous la forme d'une série de tutoriels regroupés au sein d'un ebook au format PDF, encore consultable aujourd'hui (Turkel & MacEachern, 2011). Les leçons étaient hébergées par le *Network in Canadian History & Environment* (NiCHE) sur un site WordPress.

Initialement pensée comme une introduction à la programmation Python à destination des historien·nes et autres « humanistes » avec de faibles compétences en traitement des données numériques, cette première publication devait être complétée par un autre volume, mais c'est finalement sous la forme d'un journal en libre accès que l'aventure continue en 2012, tandis que les thématiques et l'équipe éditoriale s'élargissent (Turkel, 2008). Dès lors, les bases et la philosophie ouverte du projet sont rapidement posées : une revue scientifique en libre accès immédiat (*diamond open access*), évaluation ouverte par les pairs (*open peer review*), licences libres et données ouvertes. Le travail éditorial collaboratif est centralisé sur GitHub dès 2014. La publication, d'abord proposée sur un site indépendant utilisant le système de gestion de contenu WordPress, s'oriente rapidement vers un site statique fondé sur la technologie Jekyll² et hébergé sur GitHub Pages (Lincoln et al., 2022).

Quatre ans plus tard, le projet continue de se développer avec l'élargissement de l'équipe éditoriale : initialement anglo-saxonne, elle comprend désormais une branche espagnole qui prend d'abord en charge la traduction de plusieurs leçons en anglais en 2017, puis commence à publier des leçons originales en espagnol en 2018. Face au succès de cet élargissement linguistique, les deux autres branches de la revue, française et portugaise, sont rapidement inaugurées : les premières leçons en français sont publiées en 2019, celles en portugais, en 2021.

Programming Historian est, depuis ses débuts, un projet à visée internationale porté par des volontaires. Depuis 2019, ses activités financières sont administrées par *ProgHist Limited*, une organisation de droit britannique à but non lucratif. Elle emploie actuellement une responsable de publication et une assistante de publication.

Processus éditorial

Le processus éditorial de la revue est très richement documenté dans les consignes aux auteur·es³, évaluateur·trices, traducteur·trices et rédacteur·trices, ainsi que dans un wiki au sein du dépôt GitHub **ph-submissions** de *Programming Historian*⁴.

La revue est gérée sur GitHub à l'aide de deux répertoires principaux : le dépôt **ph-submissions** pour la préparation des leçons et le dépôt **jekyll** pour la publication et la maintenance des leçons.

De manière synthétique, la chaîne éditoriale (*workflow*) de *Programming Historian* comprend sept grandes étapes :

0. Proposition (*proposal*) d'une leçon originale ou d'une traduction par la rédaction

en chef.

1. Réception de la leçon ou de la traduction par l'équipe éditoriale de la branche concernée.
2. Première lecture de la leçon par le rédacteur ou la rédactrice à qui la leçon a été attribuée (*initial edit*). Il s'agit principalement ici d'évaluer si elle est conforme aux principes d'ouverture, d'inclusivité et de durabilité de la revue, ainsi que son niveau de difficulté.
3. Première révision de la leçon par l'auteur·e (*first revision*).
4. Évaluation ouverte (*open peer review*) de la leçon à l'aide d'un ticket (*issue*) dans GitHub.
5. Deuxième révision de la leçon par l'auteur·e (*second revision*).
6. Réglages techniques pour garantir la pérennité et l'accessibilité de la leçon (*sustainability, accessibility*).
7. Mise en ligne de la leçon sur le site de *Programming Historian* (*publication*).

Ces étapes structurent la revue et définissent le rôle de chaque contributeur ou contributrice dans le processus éditorial. Elles ont été intégrées à la revue francophone, à l'instar de ses sœurs hispanophones et lusophones. Toutefois, chaque langue a évidemment ses spécificités dans les leçons publiées par *Programming Historian*.

Particularités de la branche francophone

La version francophone de *Programming Historian* a été lancée, comme nous l'évoquons, en avril 2019. Une équipe de rédaction a été constituée autour d'une première rédactrice en chef, Sofia Papastamkou. Dès son lancement, le *Programming Historian en français* a adopté des dispositions qui ont influé sur ses principes éditoriaux. En effet, dès l'origine, la revue française s'est engagée en faveur de l'utilisation de l'écriture inclusive, en suivant les principes arrêtés par l'Office québécois de la langue française (Papastamkou, 2019). Ce choix influence ainsi la manière dont les leçons sont soit traduites, soit proposées spontanément par des auteurs francophones.

Le *Programming Historian en français* a pour objectif de s'adresser à un public francophone le plus large possible. Lors de sa création, en 2019, la revue a pris forme

grâce à une première rédactrice en chef et un rédacteur dont les activités n'étaient pas localisées en France (Papastamkou, 2018). La branche francophone s'est donc construite avec des membres issus de plusieurs pays francophones et dans un contexte général d'ouverture de la science (*open science*). En 2020, une loi de programmation de la recherche est promulguée et inscrit la science ouverte dans les missions des chercheurs·euses et des enseignant·es chercheur·euses avec pour objectif affiché de rendre la totalité des publications en accès ouvert d'ici 2030 (Légifrance, 2020). Ce contexte influence donc les habitudes de travail des chercheur·euses et personnels d'appui travaillant en France, dont certain·es contribuent à la revue *Programming Historian en français*.

Actuellement, les rédacteur·trices de la branche francophone du *Programming Historian* sont tous membres du milieu universitaire — français et québécois —, plus particulièrement dans le domaine des sciences humaines et sociales, et ont divers statuts : doctorant·e, postdoctorant·e, enseignant·e chercheur·e ou ingénieur·e d'études. La symbiose de ces corps de métier est d'ailleurs une force pour faire coexister différents regards et méthodes de travail qui auraient, peut-être, plus de peine à dialoguer dans un cadre de travail plus formel. Peut-être s'agit-il là d'un héritage des THATCamp évoqués en introduction, ayant contribué à définir ce que sont les humanités numériques ? La participation à l'aventure *Programming Historian* est basée sur le volontariat, bien que cette activité puisse être soutenue institutionnellement, en allouant du temps de travail à cette activité. Bien que le constat ne soit pas figé, les personnes gravitant autour de *Programming Historian* échappent donc au profil des « petites mains », définies par Françoise Waquet comme « celles et ceux qui sont derrière ces multiples produits numériques que sont les catalogues de bibliothèques en ligne, les sites web présentant des institutions et des projets de recherche, les bases de données, les portails d'archives ouvertes », en les nommant « travailleurs du clic » (Waquet, 2022). L'auteure va même plus loin en parlant de « non-personnes invisibles » et autres « petites mains effacées » notamment composées de personnels statutaires, mais aussi d'un précaire de contractuels et de vacataires. Elle fait le parallèle, pour les humanités numériques, aux contributeur·trices précaires, d'abord apparues aux États-Unis, notamment les départements de littérature. D'ailleurs, Françoise Waquet évoque plus spécifiquement les « “petites mains” de la science ouverte », qui œuvrent à ce qu'une publication ou un jeu de données produit par un·e chercheur·se soit diffusé librement. Ce profil correspond souvent à un·e professionnel·le de l'information scientifique et technique, à des « médiateurs de la “science partagée” [qui] demeurent dans l'ombre ». Au sein de *Programming Historian*, toutes les participations sont visibles et valorisées, autant que possible, notamment en début de leçon avec la mention des personnes ayant œuvré à son édition et mise en ligne (responsable du suivi éditorial, traducteur·trice, évaluateur·trice, etc.).

Le *Programming Historian en français* a souhaité reprendre le système de sa revue mère en nommant les personnes ayant contribué à une leçon lors de chacune des étapes du processus d'édition. Ainsi, dans l'en-tête de chaque leçon, on trouve la personne ayant assuré le suivi éditorial et celles qui l'ont évaluée. Pour les traductions, on y ajoute l'auteur·e de la traduction, la personne chargée du suivi de traduction et les

évaluateur·trices de celle-ci. Face au constat de Françoise Waquet, pour lequel nous avons pris soin de nous attarder, l'équipe francophone n'échappe pas à ce contexte concernant les forces vives de la production éditoriale ouverte. Dès son lancement en 2019, le *Programming Historian en français* a pu compter sur l'appui de quelques « petites mains » ayant traduit ou relu des leçons anglophones pour enrichir la revue francophone naissante (Papastamkou, 2019). L'équipe éditoriale de la branche française s'est vite constituée et étoffée au fil des années. En 2022, la première rédactrice en chef a changé de responsabilité au sein de *Programming Historian* et a été remplacée par Marie Flesch dans l'équipe française. Cette dernière a intégré de nouveaux et nouvelles rédacteur·trices l'année suivante, favorisant ainsi l'implication de membres issus de la communauté des sciences humaines et sociales francophones.

Nous évoquerons bientôt en détail la volonté, pour la revue, de faire communauté. Pour ce qui concerne les réseaux autour desquelles elle gravite, la revue francophone peut compter sur des listes de diffusion que les membres de l'équipe ont l'habitude de consulter. Ainsi, des appels à traductions ou à leçons originales sont diffusés sur ces listes, mais aussi publiés sur le site de la revue (Flesch & Chevrie, 2024). Ces appels à traduction correspondent, le plus souvent et implicitement, à une traduction de l'anglais vers le français. Toutefois, en février 2025, un appel à traduction de l'espagnol vers le français a été publié afin de pallier le manque de leçons francophones et anglophones sur le langage de balisage XML-TEI⁵, en comparaison à nos homologues hispanophones (Flesch & Chevrie, 2025).

Enfin, il apparaît que le contexte et la langue francophones peuvent être un frein à certaines traductions. De fait, les exemples choisis par la leçon d'origine peuvent parfois sembler plus lointains. En outre, les captures d'écran proposées doivent être traduites dans la mesure du possible pour permettre une meilleure compréhension. En 2022, dans une présentation de la revue francophone au colloque Humanistica, il était déjà fait mention d'appréhender la traduction de l'anglais vers le français en marquant une distance vis-à-vis d'une langue trop technique : l'enjeu consiste à surmonter les difficultés didactiques qui en découlent, tout en préservant un français standard et compréhensible par le plus grand nombre (Gille Levenson & Patat, 2022).

Évaluation ouverte par les pairs

L'ensemble des chapitres linguistiques de *Programming Historian* fonctionne sur le modèle de l'évaluation ouverte par les pairs, expression qui désigne un ensemble de méthodes d'évaluation visant à rendre publiques les évaluations des articles et les différentes étapes et versions d'une production scientifique (Ross-Hellauer, 2017). Ces systèmes d'évaluation peuvent avoir lieu sur des infrastructures conçues *ad hoc*, comme celle de [F1000Research](#). *Programming Historian* a décidé d'adopter cette approche en quittant Wordpress pour GitHub en 2014 (McDaniel, 2014). En effet, GitHub permet — indirectement, car il ne s'agit pas d'une plateforme de publication en soi — de mettre en place un système d'évaluation ouvert, par le biais des tickets (*issues*), qui servent en temps normal à soumettre des suggestions d'amélioration d'un projet ou de

correction de bugs⁶. Dans le cas qui nous intéresse, c'est bien des commentaires sur une leçon soumise qui sont publiés dans un ticket, et qui permettent de construire un dialogue entre des évaluateur·ices et l'auteur·e afin de proposer une leçon de qualité. L'ouverture de ce mode d'évaluation porte ainsi non seulement sur les commentaires laissés par les évaluateur·trices, mais aussi sur les modifications apportées à la leçon en cours de construction : pour retracer l'histoire d'une leçon, il suffit de regarder les différentes modifications apportées via le système de révision (*commits*).

Cette méthode globale a plusieurs mérites, dont celui de rendre public le travail d'édition et de correction des leçons soumises. Du point de vue relationnel, elle permet de même d'obliger les évaluateur·trices à la modération dans leur propos et d'éviter des critiques trop abruptes ou violentes⁷.

Il est à noter que cette évaluation ouverte n'est pas entièrement obligatoire : si le système de gestion de version est appliqué à la construction de toutes les leçons, l'auteur·e d'une leçon peut décider de ne pas rendre publiques les évaluations, qui lui sont alors transmises par message privé.

II) Co-existence des temporalités et traductions face aux enjeux de pérennité

Temps moyen entre la publication d'un original et sa traduction : quelques chiffres

À l'heure actuelle, le chapitre francophone compte 35 leçons, dont 30 traductions. 7 ans s'écoulent en moyenne entre une leçon et sa traduction ; la médiane est de 8 ans. Le délai minimal est d'un an (relativement incompressible étant donné le temps de l'édition), et le délai maximal de 13 ans, ce qui correspond à la traduction des premières leçons de *Programming Historian* qui a débuté autour de 2012. Il y a donc un grand nombre d'années entre la date de publication d'une leçon et la date de sa traduction, ce qui explique la nécessité d'apporter au contenu des leçons un certain nombre de modifications, au moins du point de vue technique. Le chapitre francophone est plus récent et un peu moins actif que le chapitre hispanophone, dont l'écart moyen entre la leçon originale et sa traduction est de 4 ans, qui reste tout de même important au regard de la rapidité de l'évolution des outils techniques⁸. Cet écart explique en partie que la traduction ne soit pas qu'un simple portage d'une leçon dans une autre langue, mais souvent une adaptation à la langue cible et aux évolutions techniques qui ont eu lieu depuis la rédaction de la leçon originale, comme nous allons le voir dans la section qui suit.

La traduction entre adaptation et réécriture

La traduction des leçons n'est donc pas un simple calque des leçons existantes. Elle peut impliquer une adaptation des contenus au public visé (francophone dans le cas qui nous intéresse) et suppose souvent l'adaptation aux changements technologiques, voire des amendements pour améliorer la leçon.

Premièrement, la traduction requiert une explicitation des référents culturels pour les rendre compréhensibles pour un public qui ne les connaît pas⁹. Pour les leçons portant sur des objets textuels, pour lesquels la langue est donc particulièrement importante, il est possible d'envisager un changement complet des exemples choisis. C'est le cas pour la traduction en cours de suivi de l'outil de publication statique CETELcean (Calarco, Riande, 2012)¹⁰ : l'exemple source est une chronique qui prend pour objet une partie du Cône Sud et du Río de la Plata, *La Argentina Manuscrita*, de Ruy Díaz de Guzmán (1559-1629). La traduction de cette leçon prend pour objet le roman *Splendeurs et misères des courtisanes* de Balzac. Toutes les leçons ne voient cependant pas leurs exemples adaptés : il s'agit alors plutôt de privilégier l'explicitation des exemples quand ils ne le sont pas dans la langue originale.

L'adaptation des leçons est rendue possible par le caractère reproductible des exemples et des expériences caractéristiques obligatoire de chaque leçon soumise. N'importe qui doit être en mesure de reproduire les manipulations et traitements montrés dans la leçon. Ainsi, la modification des exemples n'a pas — en théorie — d'impact sur la répliquabilité des expériences de la leçon : la structure des leçons et la philosophie de l'ouverture qui les sous-tend favorisent en elles-mêmes leur traductibilité.

La traduction d'une leçon peut être considérée comme une forme de seconde relecture par les pairs, d'abord prise en charge par la personne effectuant la traduction. En effet, des erreurs peuvent avoir été publiées malgré la première relecture ; par ailleurs, les sensibilités diffèrent logiquement entre personnes et la personne en charge de la traduction peut considérer comme important de réorienter ou de reformuler un paragraphe considéré comme peu clair.

Le premier exemple d'adaptation que nous pouvons donner porte sur l'évolution technologique qui demande souvent des modifications dans la traduction, modifications qui seront souvent reportées dans la leçon originale. Ainsi, dans la leçon mentionnée plus haut sur CETELcean, la traductrice a pris l'initiative de changer l'éditeur de code, car l'éditeur utilisé dans la leçon originale était devenu obsolète.

Un second exemple pourrait être donné sur des modifications apportées au contenu des leçons pour le rendre plus clair pour le lectorat. Nous proposons ci-dessous une capture d'un commentaire à l'intention d'un éditeur, publié sur GitHub par une traductrice :

Tout d'abord, je précise que j'ai pu tester toutes les manipulations proposées dans mes deux environnements Linux-Ubuntu et Windows 11 sans rencontrer aucun problème. Par contre, j'aimerais savoir si je peux prendre quelques libertés avec le texte source pour :

1. ajouter **des informations utiles** pour le lectorat francophone (ex : précisions spécifiques pour la locale FR, ou bien précisions techniques pour les environnements Linux/Windows 11), soit dans le texte soit en note ;
2. ajouter une ou deux **ressources en français** à celles mentionnées en anglais au début de l'article pour l'installation de Git Bash (je pense à nos lecteurs et lectrices non anglophones) ;
3. ajouter **une précision** dans le texte, parce que j'ai moi-même été surprise lorsque je n'ai pas constaté un comportement, alors qu'il est seulement lié à l'utilisation d'un logiciel spécifique ;
4. corriger une **coquille** dans un chemin d'arborescence.

Pour vous donner quelques cas concrets, le **1er point** découle de plusieurs constats :

- L'auteur ne semble pas familier de l'environnement **Linux** et sous-entend que les chemins sont identiques sur Linux et macOS.

=> Il y a de *légères* différences, mais j'aimerais ajouter les précisions des chemins Linux pour les personnes qui n'ont pas l'habitude d'utiliser des lignes de commande.

Figure 1 : Extrait d'une correspondance entre une traductrice et un évaluateur. La traductrice suggère des modifications qui portent à la fois sur l'adaptation à un public francophone, et sur l'amélioration de la leçon

Dans ce message, la traductrice suggère à la fois des ajouts pour améliorer la lisibilité et la compréhensibilité du texte pour le public francophone, et des modifications du texte source qui contient des erreurs ou inexactitudes. Une traduction de leçon n'est donc pas la seule adaptation d'un texte, il s'agit bien d'améliorer la version originale en la corrigeant ou la complétant éventuellement. Dans ce sens, la leçon repasse au crible de l'évaluation et le traducteur fait office de nouvel évaluateur.

III) Une politique d'ouverture, de diversité et d'inclusivité

Accès ouvert et multilinguisme

Programming Historian a pour objectif de rendre ses leçons les plus ouvertes et accessibles possibles. Le choix d'une licence CC BY (*Creative Commons Attribution*) permet d'utiliser ces derniers, de les partager, de les modifier et de les diffuser, y compris à des fins commerciales, à condition de créditer l'auteur·e original·e. Le processus de publication, décrit en première partie, se veut également le plus ouvert, notamment à travers le déploiement du site et le suivi des leçons, intégralement

déposés sur GitHub. Cela permet d'exposer les dimensions sociale et technique de l'élaboration d'une leçon de façon transparente, ouverte et documentée (Gille Levenson *et al.*, 2022). En un sens, l'ouverture du suivi contribue à lutter contre les éventuelles discriminations ou inégalités d'accès. C'est dans ce suivi qu'apparaît systématiquement la politique contre le harcèlement de la revue, également publiée dans les consignes aux rédacteur-trices. Par ailleurs, dans la même optique d'accessibilité, les outils et logiciels présentés dans chaque leçon publiée sont *open source* et retenus par le comité de rédaction pour leur pérennité.

Une des spécificités du chapitre francophone de la revue *Programming historian* est un attachement particulier à la promotion d'outils et logiciels libres, ouverts et pérennes. Les leçons doivent ainsi pouvoir être reproduites sans contrainte par le lectorat (enjeu de liberté), et doivent pouvoir être suivies le plus longtemps possible (enjeu de pérennité). Quelques exceptions peuvent cependant être notées, comme la leçon « La reconnaissance automatique d'écriture à l'épreuve des langues peu dotées » (Vidal-Gorène, 2023). Cette leçon s'appuie sur une infrastructure technique d'HTR fermée et propriétaire, mais il a été jugé que la méthodologie qui y était présentée, très généraliste, et qui occupait en volume les trois quarts de la leçon pouvait être utile et servir hors de l'utilisation de la plateforme qui servait d'exemple et était promue par son auteur.

Par son multilinguisme, le *Programming Historian* a fait de la traduction l'un de ses piliers en termes d'ouverture et d'efficacité didactique, assurant un nombre de visites du site toujours plus grand et que l'on peut estimer à 1 million par an en 2023 (Crymble & Im, 2023). Ainsi, dès le lancement du *Programming Historian en español* en 2018, puis du *Programming Historian en français* en 2019, la revue s'est engagée en faveur de la diversité linguistique et géographique des humanités numériques, impliquant une meilleure appréhension des contextes institutionnels, historiques, culturels et économiques respectifs (Sichani *et al.*, 2019). Ce faisant, la revue manifeste une volonté de soutenir la diversité linguistique et de combler le manque de ressources disponibles en français pour l'apprentissage des outils numériques en sciences humaines et sociales.

Par ailleurs, d'un point de vue technique, le multilinguisme a été intégré au site statique déployé sur GitHub à travers une configuration avancée de Jekyll (Lincoln, 2020), permettant d'établir une table des traductions de chaque leçon¹¹. À partir de cette normalisation de la structure des leçons en quatre langues, il a été possible d'aligner à la phrase et de manière automatisée les leçons du catalogue de *Programming Historian* afin d'étudier les modalités de traduction des termes techniques et dans l'optique de créer un vocabulaire technique multilingue (Gille Levenson *et al.*, 2024).

Faire communauté

Cependant, l'accès ouvert au *Programming Historian* ne peut, à lui seul, favoriser la diversité et l'inclusion, d'autant plus que l'expansion d'une communauté de pratique ouverte soulève de nouvelles questions sur ces enjeux. D'abord, dans ses rangs, la

revue entend, dans l'idéal, assurer une parité de genre dans les comités de rédaction ainsi que dans le choix des relecteurs·trices de chaque leçon. En outre, les équipes de rédactions de *Programming Historian* ont à cœur de faire connaître la revue ainsi que ses objectifs. Cette publicité, au sens premier du terme, se matérialise par des communications scientifiques, centralisées sur le site de la revue¹². L'enjeu consiste à faire connaître et promouvoir ces pratiques éditoriales singulières.

De plus, pour que les leçons puissent demeurer académiques et accessibles au plus grand nombre, la revue cherche à préserver une ouverture à un lectorat composé de non-spécialistes, tout en gardant une exigence rédactionnelle de recherche. En ce sens, trois niveaux de difficulté ont été établis pour chaque leçon publiée, en fonction des opérations à réaliser : facile, moyen et avancé. Ces critères constituent une mise en garde, mais instaurent aussi une gradation de difficulté dans les leçons portant sur un outil ou un sujet similaire.

Pour assurer au lectorat un maximum de leçons en phase avec ses problématiques propres aux humanités numériques, chaque chapitre de *Programming Historian* cherche un public toujours plus étendu. Ainsi, si le chapitre hispanophone a su atteindre un lectorat sud-américain, le chapitre francophone vise un public plus large que son lectorat français. S'il s'adresse facilement à un public belge, canadien ou suisse, *Programming Historian en français* espère aussi faire communauté avec les pays du Sud composés de locuteur·trices francophones. En 2022, on recensait notamment environ 50 millions de francophones en République démocratique du Congo, presque 15 millions en Algérie, plus de 13 millions au Maroc ou encore 11 millions au Cameroun (Organisation internationale de la francophonie, 2022). *Programming Historian en français* a tout intérêt à renforcer la diversité des pratiques et problématiques de recherche en élargissant sa communauté de locuteur·trices à des pays jusqu'à présent moins représentés à travers les personnes impliquées dans la rédaction de leçons. La question des inégalités d'accès à un ordinateur et, parfois, à des ressources suffisantes pour réaliser les exercices d'une leçon peut légitimement se poser. En outre, bien que 75 % des internautes soient originaires des pays du Sud, la plupart des serveurs et des infrastructures de connaissances se trouvent dans des zones anglophones, ce qui empêche d'avoir un lien direct entre un lectorat issu des pays du Sud et les leçons francophones publiées par la revue (Isasi et al., 2023).

Traduire une leçon... et ses exemples ?

Puisque le chapitre anglophone de *Programming Historian* est le premier à avoir vu le jour, en 2008, il demeure, fort logiquement, le plus pourvu en leçon. La création des chapitres hispanophones, francophones et lusophones implique un travail de traduction des leçons déjà publiées en anglais. Le contenu peut sembler facile d'accès, car déjà passé par le circuit éditorial et mis en ligne par la revue, avec un droit d'auteur CC BY 4.0 autorisant la réutilisation. Toutefois, certains exemples de jeux de données ou de mise en contexte comptent parmi les éléments plus complexes à traduire. Récemment, en janvier 2025, la question s'est posée lors de la traduction d'une leçon portant sur les données ouvertes liées, initialement publiée en anglais en 2017 et dont la traduction est

en cours (Blaney). Dans la leçon originale, il est question de différencier deux Jack Straw homonymes grâce aux données liées par l'intermédiaire de leurs identifiants VIAF (Fichier d'autorité international virtuel). Le premier « Jack Straw » est un rebelle anglais du XIV^e siècle, tandis que le second est un important ministre britannique de l'administration de Tony Blair. La leçon montre qu'il est possible de distinguer les deux Jack Straw en localisant le lieu d'activité. En l'occurrence, l'homme politique était membre du parlement britannique, représentant de la circonscription de Blackburn. Ces référentiels culturels, s'ils s'adressent à un lectorat anglophone, ne peuvent être aussi immédiatement compréhensibles auprès d'un lectorat francophone qui, par exemple, ne ferait pas partie du Commonwealth¹³. Ce type de réflexion, aussi anecdotique soit-il, doit trouver un juste milieu entre la compréhension immédiate du lectorat et l'alignement multilingue des leçons.

Enfin, puisque chaque chapitre publie ses leçons au fil de l'eau, comment quantifier les spécificités disciplinaires ou les outils représentés par les leçons ? L'index des leçons de chaque chapitre est pourvu de deux filtres intégrés. Au-delà du nombre total de leçons publiées que nous évoquons en introduction, ces filtres permettent de distinguer certaines tendances dans les activités attribuées. D'abord, une liste fermée de cinq activités, sous forme de verbes d'action, indique à quelle étape du traitement des données la leçon se situe. Les leçons se distribuent de la façon suivante :

Nombre total de leçons	Chapitre de Programming Historian	Acquérir	Transformer	Analyser	Présenter	Préserver
35	En français	3	18	6	5	3
117	In english	13	37	37	28	2
66	En español	9	27	15	13	2
54	Em português	8	22	10	11	3

Tableau 1 : Répartition des leçons publiées par langue et par étape du traitement des données.

Dans tous les chapitres, les leçons portant sur la transformation des données sont les plus nombreuses. Cependant, le chapitre français montre une surreprésentation de cette catégorie, correspondant à la moitié de toutes les leçons publiées dans cette langue. Ce tableau témoigne aussi d'un manque d'homogénéité dans la distribution des leçons d'une étape à l'autre. Tous les chapitres ont, pour le moment, peu exploré ou rédigé de leçon en lien avec la préservation. À ce titre, le chapitre français peut s'enorgueillir d'avoir publié récemment une leçon originale sur l'archive Software Heritage (Granger *et al.*, 2024).

Le second filtre des 272 leçons correspond aux sujets renseignés dans les métadonnées de chaque leçon pour en documenter le contenu. Il y a en moyenne 1,7 sujet renseigné dans une leçon. Ces 470 sujets attribués, triés par chapitre, permettent d'apporter une analyse plus fine de la couverture thématique proposée par *Programming Historian* :

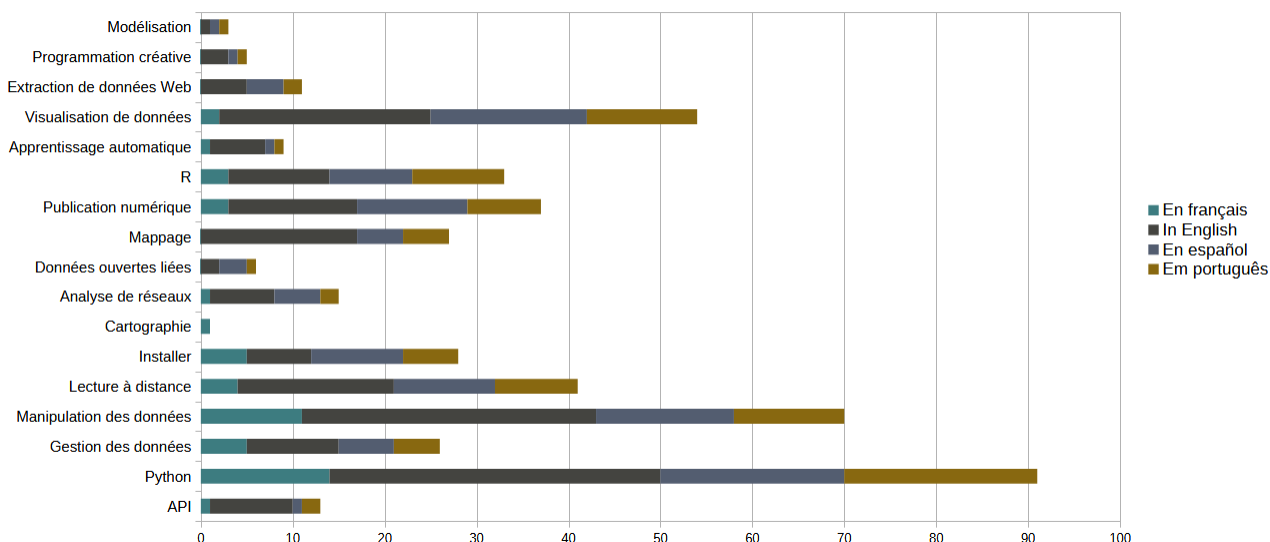


Schéma 1 : Répartition des leçons publiées par thématiques et par langue, juillet 2025
(les couleurs attribuées correspondent à l’habillage graphique de chaque chapitre linguistique)

Ces données confirment l’intérêt de *Programming Historian*, dès sa forme d’existence primitive, pour le langage de programmation Python. Par rapport à leur nombre de leçons publiées, les chapitres montrent également un vif intérêt pour les leçons portant sur la « manipulation des données », terme dont l’acception est suffisamment large pour couvrir de nombreuses thématiques. En revanche, le chapitre français semble pauvre en leçons traitant de la « visualisation des données » (2), en comparaison aux autres chapitres. Certaines thématiques ne sont, pour le moment, que très peu couvertes par la revue, à l’image des données ouvertes liées, de la modélisation ou encore de la programmation créative. À première vue, une seule leçon porte sur la cartographie. Cette dernière, publiée par le chapitre français, propose une prise en main introductive de l’outil Heurist (Paillusson, 2022). Pourtant, le terme de mappage (ou *mapping*), utilisé par les autres chapitres, semble plus englobant, mais demeure approprié. Enfin, parmi les thématiques émergentes de la revue figure l’extraction de données web (*Web Scraping*). Le chapitre français n’a, pour le moment, pas de leçon publiée sur ce thème. Ce graphique constitue un état de l’art de l’existant et amène *Programming Historian* à réfléchir sur la manière dont les indicateurs doivent être conçus pour être mieux à même de traduire la réalité des thématiques couvertes.

Conclusion

Cet état des lieux de *Programming Historian en français* nous permet de mettre en perspective l’historique naissant de ce chapitre avec ses ambitions actuelles. Issu d’un projet universitaire, canadien et anglophone, *Programming Historian* a su prendre vie sous la forme d’une revue, s’adressant en priorité à un public novice sur le traitement des données en sciences humaines et sociales grâce à des outils numériques. De ce grand principe est né d’un processus éditorial ainsi qu’un site rapidement déployé sur *GitHub Pages* et sur lequel les leçons sont publiées. Ce fonctionnement, répliquable et répliqué, a permis à trois autres langues d’intégrer la revue.

Le multilinguisme et, à travers lui, la traduction sont à la fois une force et une faiblesse pour les chapitres hispanophone, francophone et lusophone. Chaque traduction implique une interrogation sur l'adaptation du contenu pédagogique et des exemples à des sphères culturelles différentes. Dans un monde où le Web propose de tutoriels majoritairement anglophones, *Programming Historian* entend diversifier son offre de leçons, tutoriels « améliorés », pour être le plus en phase avec les besoins d'une communauté à construire, dont les valeurs poussent à mettre en avant les logiciels libres et gratuits. Ces leçons sont conçues comme des articles scientifiques, c'est pourquoi l'enjeu principal de la revue concerne sa pérennité. D'un point de vue technologique, la plateforme dépend d'une solution poussée au maximum de ses capacités, ce qui la pousse à s'interroger et d'envisager actuellement une migration vers une autre technologie. Pour ce qui concerne le contenu, la pérennité des leçons passe par leur pertinence : un outil pris pour exemple peut ne plus être accessible, un logiciel peut tomber en désuétude au bout de quelques années. Ces risques sont aussi des forces pour l'équipe de la revue, dont les compétences nourrissent les réflexions sur l'adaptabilité de *Programming Historian* sur le long terme, après une décennie sur GitHub.

Parmi les perspectives les plus structurantes incarnées par le projet *Programming Historian*, il y a celle de faire vivre la revue autour d'une communauté de pratiques. L'objectif est de faire fructifier ce réseau de contributeurs et contributrices afin de renouveler le contenu des leçons au gré des besoins. Ces derniers sont parfois difficilement identifiables, c'est pourquoi la revue est sensible aux enjeux d'inclusivité et de diversité. Le suivi éditorial est ouvert et chaque personne impliquée dans la publication d'une leçon doit voir sa contribution valorisée et réalisée dans un cadre de confiance.

Depuis 2008, la revue *Programming Historian* ne propose plus seulement des leçons sur la programmation. De même, son lectorat n'est plus seulement constitué d'historiens et d'historiennes : la revue s'adresse à l'ensemble des personnes qui travaillent ou étudient dans le domaine des sciences humaines et sociales. Cette montée en puissance a été permise par un déploiement solide de la revue, un accès libre et gratuit aux leçons, ainsi qu'un fonctionnement reproductible.

Notes

¹“Nothing exists at present for the historian who wants to use a computer but has not the slightest idea how to proceed” (Shorter, 1971: 1).

²Jekyll. (2025, janvier 29). *Jekyll. Simple, blog-aware, static sites*. Jekyll. <https://jekyllrb.com/>

³*Programming Historian*. (2025, mai 1). *Contribuer*. <https://programminghistorian.org/fr/contribuer>

4 *Programming Historian*. (2025, avril 3). *Publishing Workflow*. GitHub. <https://github.com/programminghistorian/ph-submissions/wiki/Home>

5 Le XML-TEI (*Text Encoding Initiative*) est une norme d'encodage de textes fondée sur le langage balisé XML, destinée à structurer, décrire et échanger des documents textuels de manière interoperable. Il est principalement utilisé dans le domaine des sciences humaines et sociales.

6 Ce système n'est pas inscrit dans le marbre et sera amené à évoluer avec la propre évolution de l'infrastructure de *Programming Historian* : des discussions sont en cours pour s'arrimer à des infrastructures éditoriales professionnelles.

7 Il existe par ailleurs un rôle de médiateur·trice dans la revue, afin de gérer tout conflit qui pourrait survenir.

8 Données produites à partir des métadonnées des leçons sur la branche gh-pages du dépôt Jekyll, dernière révision 7cd5d2b2a533. Il est à prévoir que l'écart se réduira au fur-et-à-mesure des années, le chapitre francophone étant né plusieurs années après l'apparition de la revue anglophone et ayant nécessité d'abord la traduction des leçons les plus simples de la revue. La production de ces statistiques est grandement facilitée par un système ouvert et structuré de publication de métadonnées. Plus généralement, la structure actuelle de la revue, composée de leçons structurées en markdown et aux métadonnées directement interrogeables par la machine – car elles sont en YAML – permet de produire des études sur les leçons en elles-mêmes. Le corpus, très riche, permet d'avoir une vision, même biaisée, des intérêts d'une communauté constituée et peut servir en lui-même à produire de la connaissance.

9 *Programming Historian, Consignes aux auteur(e)s*, 10 janvier 2025 : <https://programminghistorian.org/fr/consignes-auteurs>

10 Calarco, G., & Riande, G. del R. (2025). Introduction à la publication web de fichiers TEI avec CETEIcean (brouillon de la traduction française). *Programming Historian*. <https://web.archive.org/web/20250617111458/https://programminghistorian.github.io/ph-submissions/fr/en-cours/traductions/publier-archives-tei-ceteicean>

11 *Programming Historian*. (2021, janvier 11). *Translation Concordance*. <https://programminghistorian.org/translation-concordance>

12 *Programming Historian*. (2025, juillet 16). *Production académique autour du projet*. <https://programminghistorian.org/fr/recherche>

13 Lien vers le ticket GitHub, traduction de la leçon "Introduction aux principes des données ouvertes liées". <https://github.com/programminghistorian/ph-submissions/issues/643>

Bibliographie

Blaney, J. (2017). Introduction to the Principles of Linked Open Data. Programming Historian. <https://doi.org/10.46430/phen0068>

Calarco, G., & Riande, G. del R. (2021). Introducción a la publicación web de archivos TEI con CETELcean. Programming Historian. <https://doi.org/10.46430/phes0056>

Calarco, G., & Riande, G. del R. (2025). Introduction à la publication web de fichiers TEI avec CETELcean (brouillon). Programming Historian. <https://web.archive.org/web/20250617111458/https://programminghistorian.github.io/p-h-submissions/fr/en-cours/traductions/publier-archives-tei-ceteicean>

Clavert, F. (2010, avril 26). THATCamp Paris ! (EN) - THATCamp Paris. <https://doi.org/10.58079/uo1a>

Crymble, A. (2021). Technology and the historian : Transformations in the digital age / Adam Crymble. University of Illinois Press.

Crymble, A., & Im, C. M. H. (2023). Measuring digital humanities learning requirements in Spanish & English-speaking practitioner communities. International Journal of Digital Humanities, 5(2), 253-282. <https://doi.org/10.1007/s42803-023-00066-x>

Dacos, M., & Mounier, P. (2015). Humanités numériques (p. 9782354761097) [Research Report]. Institut français. <https://hal.science/hal-01228945>

Fiormonte, D. (2021). Taxation against Overrepresentation ? The Consequences of Monolingualism for Digital Humanities. In Alternative Histories of the Digital Humanities (Dorothy Kim and Adeline Koh, p. 333-376). Punctum Books.

Flesch, M., & Chevrie, C. (2024, juillet 4). Appel à propositions (édition française). <https://programminghistorian.org/posts/appel-a-propositions>

Flesch, M., & Chevrie, C. (2025, février 28). Appel à traductions de l'espagnol vers le français : Leçons sur la TEI. <https://programminghistorian.org/posts/appel-a-traductions>

Gille Levenson, M., Papastamkou, S., & Ringwald, C. (2022, juin 22). Programming Historian : Un lieu de collaborations et d'interactions multiples. DHNord 2022 : Travailler en humanités numériques: collaborations, complémentarités et tensions, Lille.

Gille Levenson, M., & Patat, G. (2022, mai). Programming Historian en français : Faire communauté pour le partage de ressources éducatives libres sur les méthodes numériques en sciences humaines et sociales francophones. Colloque Humanistica 2022. <https://hal.science/hal-03672420>

Gille Levenson, M., Ringwald, C., Flesch, M., Isasi, J., Papastamkou, S., Quiroga, R., & Valentine, D. (2024, mai). Connecter les chapitres linguistiques de Programming Historian ? HAL. <https://hal.science/hal-04557817>

Granger, S., Mèlès, B., & Santos, F. (2024). Préserver et rendre identifiables les logiciels de recherche avec Software Heritage. The programming historian en français, 6. <https://doi.org/10.46430/phfr0034>

Isasi, J., Quiroga, R., Siddiqui, N., Paulino, J. V., & Wermer-Colan, A. (2023). A Model for Multilingual and Multicultural Digital Scholarship Methods Publishing. In The Case of Programming Historian (SCOPUS:85180916224; p. 17-30). Taylor and Francis. <https://doi.org/10.4324/9781003393696-3>

Jekyll. (2025, janvier 29). Jekyll. Simple, blog-aware, static sites. Jekyll • Simple, Blog-Aware, Static Sites. <https://jekyllrb.com/>

Légifrance. (2020, décembre 30). LOI n° 2020-1674 du 24 décembre 2020 de programmation de la recherche pour les années 2021 à 2030 et portant diverses dispositions relatives à la recherche et à l'enseignement supérieur—Dossiers législatifs—Légifrance. <https://www.legifrance.gouv.fr/dossierlegislatif/JORFDOLE000042137953/>

Lincoln, M. D. (2020, mars 1). Multilingual Jekyll : How The Programming Historian Does That. Matthew Lincoln. <https://matthewlincoln.net/2020/03/01/multilingual-jekyll.html>

Lincoln, M., Isasi, J., Melton, S., & Laramée, F. D. (2022). Relocating Complexity : The Programming Historian and Multilingual Static Site Generation. Digital Humanities Quarterly, 016(2).

Longhi, J. (2019). Contours, perspectives et tensions des « humanités numériques ». Sens-Dessous, N°24(2), 123. <https://doi.org/10.3917/sdes.024.0123>

McDaniel, C. (2014, novembre 5). How We Moved the Programming Historian to GitHub Pages. <https://programminghistorian.org/posts/how-we-moved-to-github>

McGrath, J. (2020, mars 3). THATCamp Reflections: On the Unfinished Business of Unconferences | THATCamp Retrospective. <https://retrospective.thatcamp.org/2020/03/03/thatcamp-reflections-on-the-unfinished-business-of-unconferences/index.html>

Organisation internationale de la francophonie. (2022). La langue française dans le monde. https://www.francophonie.org/sites/default/files/2023-03/Rapport-La-langue-francaise-dans-le-monde_VF-2022.pdf

Papastamkou, S. (2018, décembre 5). François Dominic Laramée Joins the Programming Historian Project Team. <https://programminghistorian.org/posts/fd-laramee>

Papastamkou, S. (2019, avril 8). Bienvenue au Programming Historian en français! <https://programminghistorian.org/posts/bienvenue-ph-fr>

Quiroga, R., Hawes, A., Sichani, A.-M., & Chevrie, C. (2024). Sustainable Growth of Multilingual Open Publishing Projects : The Case of Programming Historian. *The Journal of Electronic Publishing*, 27(1), Article 1. <https://doi.org/10.3998/jep.5571>

Ross-Hellauer, T. (2017). What is open peer review ? A systematic review. *F1000Research*, 6, 588. <https://doi.org/10.12688/f1000research.11369.2>

Shorter, E. (1971). *The Historian and the Computer : A Practical Guide*. Prentice-Hall.

Sichani, A.-M., Baker, J., Afanador Llach, M. J., & Walsh, B. (2019). Diversity and inclusion in digital scholarship and pedagogy : The case of The Programming Historian. *Insights: the UKSG journal*. <https://doi.org/10.1629/uksg.465>

Terras, M. (2011). *Quantifying Digital Humanities*. UCL Centre for Digital Humanities. <https://web.archive.org/web/20250629045645/https://www.ucl.ac.uk/infostudies/melissa-terras/DigitalHumanitiesInfographic.pdf>

Thély, N., Serres, A., & Deuff, O. L. (2014). *Le THATCamp comme nouvelle forme de communication scientifique?* Editions de la Maison des Sciences de l'Homme. <https://doi.org/10.4000/books.editionsmsmsh.2183>

Turkel, W. J. (2008, janvier 14). *Digital History Hacks (2005-08) : The Programming Historian*. *Digital History Hacks (2005-08)*. <https://digitalhistoryhacks.blogspot.com/2008/01/programming-historian.html>

Turkel, W. J., & MacEachern, A. (2011, avril 9). *The Programming Historian [Text]*. NiCHE: Network in Canadian History & Environment (2007-11). <https://niche-canada.org/research/niche-digital-infrastructure-project/the-programming-historian/>

Vidal-Gorène, C. (2023). La reconnaissance automatique d'écriture à l'épreuve des langues peu dotées. *Programming Historian*. <https://doi.org/10.46430/phfr0023>

Waquet, F. (2022). *Dans les coulisses de la science : Techniciens, petites mains et autres travailleurs invisibles*. CNRS éditions.