

HNU2000-A25

Humanités Numériques : technologies

[**Mathilde Verstraete**](#)

Automne 2025

Plan du cours

- *Sigle du cours* : HNU2000
- *Titre du cours* : Humanités numériques : technologies
- *Trimestre* : Automne 2025
- *Titulaire du cours* : Mathilde Verstraete
- *Coordonnées* : mathilde.verstraete@umontreal.ca
- *Horaire* : Mardi 08h30-11h30 (du 02/09 au 09/12/2025)
- *Lieu* : Cf. votre centre étudiant

Description du cours

Espace de formation pratique basée sur des compétences pour les méthodes de base dans les humanités numériques, y compris l'exploration de textes, l'analyse de données, l'analyse du web et les systèmes d'information géographique.

Objectifs et contenu du cours

Le numérique habite l'ensemble de nos vies et touche aussi, et surtout, à nos activités purement « humanistes », ou même « humaines ».

([Sinatra & Vitali-Rosati, 2014](#))

Les disciplines humanistes au sens large s'appuient de plus en plus sur des outils informatiques pour explorer, traiter, analyser, diffuser leurs objets d'étude. Cette irruption des outils issus des sciences dites dures dans les sciences humaines a donné naissance à un nouveau champ interdisciplinaire, celui dit des *humanités numériques*.

L'objectif principal du cours **HNU2000 – Humanités numériques : technologies** est de familiariser les étudiant·e·s aux principaux outils, technologies, logiciels, utilisés dans ce champ. Cette familiarisation passera par la manipulation concrète d'outils numériques et par la réalisation de projets pratiques.

Parmi les thématiques abordées durant le cours:

- La connaissance de son ordinateur ;
- L'encodage et la structure de fichiers numériques ;

- Les formats d'écriture ;
- Le versionnement de ses données ;
- Le traitement et la gestion des références bibliographiques ;
- La fouille et l'analyse de textes ;
- La reconnaissance automatique d'écritures manuscrites ;
- etc.

Les « humanités numériques » sont à la fois une méthode scientifique, un programme de recherche ou une approche pluridisciplinaire : elles offrent de nouvelles perspectives pour appréhender, lire et comprendre le monde qui nous entoure. Depuis l'avènement du numérique, la compréhension des écosystèmes technologiques devient une nécessité dans le champ des sciences humaines. Le cours HNU2000 « Humanités numériques : technologies » est une opportunité pour explorer de façon originale les outils théoriques et pratiques utilisés dans les humanités numériques : lire, écrire, chercher, explorer, visualiser, analyser, publier, etc. Ce cours est un espace de formation pratique et de découverte des méthodes de base en humanités numériques, avec une orientation vers les démarches d'écriture, d'édition et de publication.

Ce cours est une initiation aux technologies utilisées dans les humanités numériques, les étudiant·e·s devront comprendre, explorer, manipuler et expérimenter des outils, des logiciels, des méthodes et des programmes informatiques. Le cours HNU1000 « Humanités numériques : théories » est un compagnon adéquat pour ce cours pratique.

À l'issue du cours les étudiant·e·s seront en mesure de comprendre les enjeux technologiques des humanités numériques, de réutiliser des concepts liés aux humanités numériques, d'appréhender des méthodes utilisées dans les humanités numériques et d'utiliser une multitude d'outils/applications.

L'outillage est souvent un aspect mal considéré en humanités, pourtant c'est la condition même de la possibilité de nos recherches : nos outils de collecte, d'enregistrement, de traitement, de prise de notes, de rédaction et de publication sont ce par quoi tout commence.

Organisation du cours

Le cours alterne entre théorie et pratique. De manière générale, chaque cours sera l'occasion d'étudier un outil. À cette fin, il sera demandé à l'étudiant·e de préparer la séance par une ou des lecture(s) et/ou des manipulations sur son ordinateur personnel (installation de logiciel, création de compte, etc.). Le cours sera l'occasion d'étudier – d'un point de vue pratique, mais aussi théorique – l'outil en question. Il pourra être demandé à l'étudiant·e de terminer des manipulations après la séance de cours et pour le cours suivant.

Dans le [calendrier](#), vous trouverez sous chaque séance ce qui sera abordé, les lectures et actions (téléchargement, création de compte, etc.) à faire avant la séance ainsi que des lectures suggérées.

! Important

Il est impératif que l'étudiant·e possède un ordinateur et l'apporte en cours.

Modalités d'évaluation

! Important

Rappel : 1h de cours = 1,5h de travail hors cours

- Participation¹ & Implication dans les exercices et travaux non évalués : 25% ;
- **28/10 Examen de mi-session** : 25% ;
- **28/10 Travail bonus** : Apprenez-moi quelque chose ;
- **09/12 Travail final** : 25% ;
- **16/12 Présentation finale** : 25%.

Examen de mi-session

- *Modalités* : examen sur table ;
- *Durée* : environ 2h ;
- *Forme* :
 - Questions à choix multiple ;
 - Questions ouvertes ;
- *Contenu* : questions théoriques portant sur :
 - Le cours ;
 - Les lectures obligatoires.

Travail bonus : *Apprenez-moi quelque chose*

- Ce travail n'est pas obligatoire :
 - Si vous ne le rendez pas → aucune pénalité ;
 - Si vous le rendez → possibilité de gagner des points supplémentaires sur la note globale du cours.
- Consignes :
 - Longueur : entre 1.5 et 3 pages ;
 - Sujet : description d'un outil, projet ou actualité de votre choix, en lien avec les humanités numériques ;
 - Contraintes techniques : doit être rédigé avec Stylo ou LaTeX ;
- Éléments évalués :

¹Par participation, j'entends la préparation, la concentration lors des séances, l'écoute active, l'implication lors des échanges.

- Qualité de la présentation générale (orthographe, structure, mise en page, ajout de liens, bibliographie minimale) ;
 - Clarté et pertinence de la description du projet choisi ;
 - Force de l'argumentation qui doit donner envie d'en savoir plus ;
- Impact sur la note finale :
 - En fonction de la qualité du travail, votre note peut monter de deux niveaux (ex. : un B+ peut devenir un A- ou un A) ;
 - Date de remise : 28 octobre (aucun délai ne sera accordé).

Examen final

Travail écrit

- *Date de remise* : au plus tard le lundi 9 décembre ;
 - Une pénalité de 10% sera appliquée par jour de retard ;
- *Format attendu* :
 - Analyse d'un outil présenté en classe ou d'un autre outil (avec approbation préalable de la chargée de cours) ;
 - Présentation soignée : orthographe, mise en page, ajout éventuel d'images, bibliographie, liens utiles ;
 - PDF, HTML ou autre – soyez créatifs (évitez les traitements de texte!) ;
 - Environ 5 à 7 pages (hors bibliographie) ;
- *Contenu attendu* :
 - Présentation de l'outil (objectifs, fonctionnement, contexte d'utilisation) ;
 - Analyse critique (points forts, limites, potentiel pour les humanités numériques) ;
 - Éventuelle mise en dialogue avec des articles évoquant l'outil en question ;
 - Mise en perspective par rapport aux thématiques du cours.

Présentation orale et discussion en classe

- *Date* : lundi 16 décembre ;
- *Durée* : environ 10 minutes de présentation + 5 minutes de discussion ;
- *Contenu attendu* :
 - Présentation synthétique de l'outil analysé dans votre travail écrit ;
 - Mise en avant des aspects les plus intéressants pour susciter l'échange avec vos collègues ;
 - Ouverture à la discussion : préparez une ou deux questions pour la classe.

Calendrier des séances et des lectures

Le cours de cette année sera axé sur le cycle de vie du texte savant en contexte numérique. Nous aborderons le texte source (séances 2-6), son analyse (8-10) et sa publication et diffusion (11-13).

Pour une description détaillée des séances, lectures obligatoires et conseillées, cf. [le calendrier des séances](#).

Ressources utilisées dans le cadre de ce cours

- Le présent site/syllabus contient l'ensemble des informations dont vous aurez besoin pour ce cours.
- Lorsque les références sont disponibles en ligne le lien est indiqué sur le site du cours. Attention, certaines ressources en ligne nécessitent une connexion UdeM : soit depuis l'Université, soit en passant par le VPN.

Plagiat et fraude

« Tout acte de plagiat, fraude, copiage, tricherie ou falsification de document commis par une étudiante, un étudiant, de même que toute participation à ces actes ou tentative de les commettre, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation ou dans toute autre circonstance, constituent une infraction. Vous trouverez à l'adresse suivante les différentes formes de fraude et de plagiat ainsi que les sanctions prévues par l'Université : <https://integrite.umontreal.ca> »

Crédits

Le contenu de ce cours doit beaucoup aux préparations réalisées par Antoine Fauchié, Margot Mellet, Alix Chagué ainsi qu'aux séances de [Débogue tes humanités](#) et à l'ouvrage *Vade-mecum informatique pour les lettres et sciences humaines* ([Debouy, 2025](#)).

Le support du cours a été créé avec [Quarto](#).

License

Tous les contenus de ce site ou de ce document sont sous licence CC BY-NC-SA : Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions.

Calendrier des séances et des lectures

[Séance 1 : 02/09] Introduction au cours et aux Humanités Numériques

- Présentation de l'enseignante et des étudiant·e·s ;
- Présentation du cours : déroulement, courte présentation des séances, participation, évaluations ;
- Introduction aux Humanités Numériques.

💡 Lectures supplémentaires

- Dacos & Mounier ([2015](#))
- Dacos ([2011](#))
- Burdick et al. ([2012](#))
- Burnard ([2012](#))
- Underwood ([2018](#))

[Séance 2 : 09/09] COMPRENDRE son environnement

- Prise en main de son ordinateur : fichiers, terminal, ligne de commande ;
- Bash ;
- L'éditeur *vs* le traitement de texte.

! Lectures obligatoires

- Sinatra & Vitali-Rosati ([2014](#))

! Outils à télécharger

- <https://vscodium.com/#install>.

💡 Lectures supplémentaires

- Schmidt ([2016](#))

[Séance 3 : 16/09] CHERCHER : corpus et ressources numériques

i Note

[17/09] Date limite pour la modification des choix de cours.

- Internet et le Web ;
- Les navigateurs et moteurs de recherche ;
- Une recherche efficace ;
- Manipulation : Isidore.

! Lectures obligatoires

- France Inter ([s. d.](#)) (podcast)
- Huma-Num ([2021](#))

! Outils à télécharger

- Créez-vous un compte [HumanID](#)

? Lectures supplémentaires

- Cardon ([2013](#))
- Larsonneur ([2008](#)), sur la question des navigateurs et moteurs de recherche, notamment
- Ochigame & Ye ([2021](#)), sur les biais géographiques des résultats aux recherches sur Google

[Séance 4 : 23/09] ORGANISER & BIBLIOGRAPHIER : métadonnées et formats

- Les métadonnées ;
- Les différents formats textuels ;
- La bibliographie : définition, intérêts, mode d'emploi ;
- Manipulation : Zotero.

! Lectures obligatoires

- Collectif ([2018](#))

! Outils à télécharger

- Téléchargez [Zotero](#) (Zotero & l'extension *Zotero Connector*) ;
- [Créez-vous un compte Zotero](#)

💡 Lectures supplémentaires

- Schrijver ([2017](#))
- Daniel ([2022](#))

[30/09]

i Note

Congé universitaire – Journée nationale de la vérité et de la réconciliation.

[Séance 5 : 07/10] REPRÉSENTER le texte (I) : formats légers

- L'écriture numérique ;
- Les formats légers : .txt, .md, .html ;
- Manipulation : [Stylo](#).

! Lectures obligatoires

- Vitali Rosati et al. ([2020](#))

! Outils à télécharger

- Créez-vous un compte/connectez-vous à [Stylo](#) via votre HumanID (cf. la Séance 3)

💡 Lectures supplémentaires

- Vitali-Rosati ([2018](#))

[Séance 6 : 14/10] REPRÉSENTER le texte (II) : formats riches et normés

Note

[07/11] Date limite pour l'abandon d'un cours.

- Les langages interprétés : l'exemple de LaTeX ;
- Introduction à la TEI ;
- Manipulations :
 - [LaTeX \(via Overleaf\)](#) ;
 - [Écrire en TEI-XML \(via LeafWriter\)](#) ;
 - [Pandoc](#) – si le temps le permet.

Lectures obligatoires

- Coombs et al. ([1987](#))

Outils à télécharger

- Créez-vous un compte [Overleaf](#)
- Téléchargez [Pandoc](#) : Nous le ferons ensemble en classe

Lectures supplémentaires

- Burnard ([2015](#)), lecture vivement conseillée
- « Just What Is TeX? » ([2023](#)), lecture vivement conseillée
- André ([1990](#))

[21/10]

Note

Semaine de lecture – pas de cours.

[Séance 7 : 28/10] EXAMEN

! Important

- Examen de mi-session ;
- Remise du travail facultatif.

[Séance 8 : 04/11] ANNOTER : des *marginalia* à l'annotation numérique

- Historique ;
- Le Web : enjeux collaboratifs ;
- Manipulation : [Hypothes.is](#).

! Lectures obligatoires

- Fredriksson & Sauret ([2019](#))

! Outils à télécharger

- Créez-vous un compte [Hypothes.is](#)

💡 Lectures supplémentaires

- Jahjah ([2017](#))

[Séance 9 : 11/11] ÉCRIRE & PUBLIER

- Le *Single Source Publishing* ;
- Manipulation : Quarto ;
- La gestion de versions ;
- Manipulation : Git et GitHub.

! Lectures obligatoires

- Fauchié & Audin ([2023](#))

! Outils à télécharger

- Créez-vous un compte [GitHub](#)

💡 Lectures supplémentaires

- Turbiau ([2023](#))

[Séance 10 : 18/11] NETTOYER & EXPLOITER les données textuelles

⚠ Avertissement

- Conférence & Workshop, A. Chagué, « De l'OCR à l'HTR : panorama général, suivi d'un workshop introductif à eScriptorium » – À confirmer !

- La notion de *data* ;
- La numérisation des corpus : qui, quoi, comment ? ;
- Une petite histoire de l'OCR à l'HTR.

❗ Lectures obligatoires

- Bermès ([2023](#))

[Séance 11 : 25/11] ANALYSER : textométrie, stylométrie et TAL

- Humanités Numériques et Texte ;
- Manipulations :
 - [Ngram Viewer](#) ;
 - [Voyant Tools](#).

! Lectures obligatoires

- Reboul & Gefen ([2019](#))

💡 Lectures supplémentaires

- Bot ([2012](#))

[Séance 12 : 02/12] VISUALISER : vers une autre lecture

- Les graphes : quelques éléments théoriques ;
- Manipulations :
 - [TimeLineJS](#) ;
 - [StoryMapJS](#).

! Lectures obligatoires

- Robichaud et al. ([2019](#))

💡 Lectures supplémentaires

- Perret ([2022a](#))
- Perret ([2022b](#))
- Drucker ([2011](#))

[Séance 13 : 09/12] IA

! Important

- Remise de l'examen final (écrit).

! Important

Cette séance est remplacée par les deux *Ateliers IA pour les SHS* :

- jeudi 11 septembre 2025 15:30 : Introduction : comment distinguer l'IA de ChatGPT ;
- jeudi 09 octobre 2025 15:30 : IA et la correction textuelle automatique : quels outils et quelles limites ?

💡 Lectures supplémentaires

- Chagué ([2022](#))
- Smits & Wevers ([2023](#))

[Séance 14 : 16/12]

! Important

[Examen final.](#)

Introduction au cours

Bienvenue !

Cette séance est dédiée d'une part à la présentation du cours, son déroulement et autres détails utiles, d'autre part à un (bref) panorama des Humanités Numériques.

Tour de table

Quelques questions :

- Quel est votre prénom ?
- Quel est votre parcours académique ?
- Qu'attendez-vous de ce cours ?
- Que savez-vous des Humanités Numériques ?

Déroulement du cours

- Cours en présentiel = présence requise ;
- Lectures obligatoires avant chaque séance :
 - Les lectures qui ne sont pas encore précisées le seront au moins une semaine avant la semaine concernée ;
 - Les lectures obligatoires serviront de base à des discussions pendant le cours et sont à connaître pour les divers examens ;
- Support/ressources disponibles à la fin de chaque cours (sur ce site web) :
 - Ce qui ne dispense aucunement la prise de note durant le cours ;
- Pendant les séances : partie théorique, manipulations, échanges.

Objectifs du cours

Cf. le [Plan de cours](#).

Présentation des séances

Cf. le [Plan de cours](#).

Modalités d'évaluation

Cf. le [Plan de cours](#).

[Séance 1 : 02/09] Introduction aux Humanités Numériques

Définitions

Comment définiriez-vous, en vos mots, les *Humanités Numériques*?

D'une part, les humanités numériques pourraient être définies comme l'application d'une méthode d'analyse informatique aux sciences humaines. En d'autres mots, l'approche des DH consiste à prendre en compte le fait que la puissance ne doit pas être limitée aux sciences dures, mais peut et doit aussi être employée pour des recherches en sciences humaines. D'autre part, les humanités numériques transcendent cet aspect technique et peuvent être pensées comme un regard global posé sur les changements culturels déterminés par le numérique ; en ce sens, les humanités numériques pourraient conduire à une sorte d'« humanisme numérique ». ([Sinatra & Vitali-Rosati, 2014](#))

Domaine de recherche et d'enseignement au croisement de l'informatique et des lettres, des arts, des sciences humaines et des sciences sociales, visant à produire et à partager des savoirs, des méthodes et de nouveaux objets de connaissance à partir d'un corpus de données numériques.² ([Commission d'enrichissement de la langue française, 2019](#))

- *Quelques* autres définitions [ici](#) ou [ici](#).

Humanités...

- Traditionnellement : lettres classiques ;
- Aujourd'hui (et surtout en Amérique du nord) : les sciences humaines ;
- Littérature, philosophie, histoire, arts vivants, linguistiques, etc.

²Cette définition a été réutilisée [sur le site de l'OQLF](#).

...Numérique(s)

- Représentation par nombre ;
- Discrétisation ;
- Numérisation du monde ;
- Culture numérique.

Pour en savoir un peu plus sur l'encodage des caractères :

https://www.youtube.com/watch?v=MijmeoH9LT4&list=PLzH6n4zXuckqmf_xUcvU5caZVoctP2ehL

Les trois moments des Humanités Numériques

Lou Burnard (2012) distingue trois moments des humanités numériques :

env. 1970 : *Literary and Linguistic Computing*

- Déjà vers la fin des années 1940 avec l'*Index Thomisticus* de R. Busa ;
- Puissance de calcul des ordinateurs afin d'*automatiser* la création d'index, le repérage de concordances, le calcul de fréquences ... ;
- Ex. : *British National Corpus*, *Thesaurus Linguae Graecae*.

env. 1980 : L'*humanities computing*

- Compréhension et maîtrise du programme informatique qui devient à proprement parler l'instrument d'une méthode de recherche et ce, dans une perspective interdisciplinaire ;
- Ex. : la TEI (*Text Encoding Initiative*).

env. 1990 : *Digital Humanities*

- Apparition du Web → disponibilité des corpus numériques → nécessité de penser des interfaces pour consulter les corpus. Nouveaux rythmes de publication, modes de diffusions, formats.
- « On est passés de l'informatique comme outil au service de la recherche au numérique comme environnement global au sein duquel s'effectue la recherche. » (Debouy, 2025, p. 15)
- Ex. : la *William Blake Archive*.

Quelques repères historiques

- Aujourd’hui, l’informatique est partout → gain de temps, facilité de traitement ; accès aux informations ;
- Nouvelles questions : où sont stockées les données ? Quelle confidentialité ? Quelle est notre relation aux outils que nous utilisons tous les jours ?
- « Le citoyen du XXI^e siècle en est bien souvent réduit à être de plus en plus un consommateur et non un utilisateur averti et libre de ces technologies numériques. » ([Debouy, 2025, p. 17](#))

Les prémisses mécaniques

- II^e s. av. J.-C. : le mécanisme d’Anticythère peut être considéré comme le premier calculateur analogique (connu). Il permettait de modéliser la course des astres grâce à une trentaine d’engrenages. Il faudra attendre près d’un millénaire pour les prochains systèmes comparables, dans les horloges de Moyen Âge. La reconstruction de ce mécanisme intéresse toujours les chercheurs aujourd’hui.



Figure 1: Le fragment principal de la machine d’Anticythère

- Env. 1640 : la *Pascaline* de Blaise Pascal, calculatrice mécanique capable d’additionner et de soustraire, conçue pour assister son père percepteur d’impôts.



Figure 2: Pascaline, conservée au musée des Arts et Métiers, Paris

- XVII^e s. – fin du XX^e s. : la règle à calcul, un instrument mécanique utilisé massivement jusqu’aux années 1970 pour faire des opérations arithmétiques et trigonométriques. Elle sera préférée à la Pascaline notamment en raison de son prix, sa facilité d’utilisation et de fabrication.



Figure 3: Règle à calcul scolaire, 10 pouces (Pickett N902-T simplex trig).

- 1801 : le métier à tisser de Jacquard, système mécanique programmable par l’utilisation de cartes perforées, introduit l’idée d’un programme externe pour automatiser des opérations.

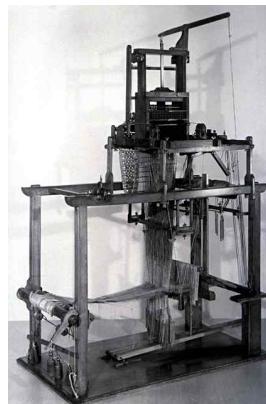
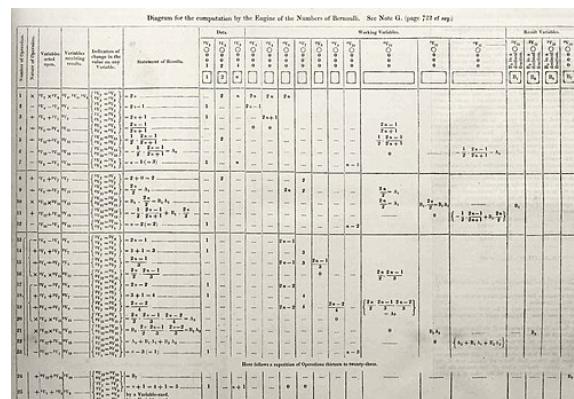
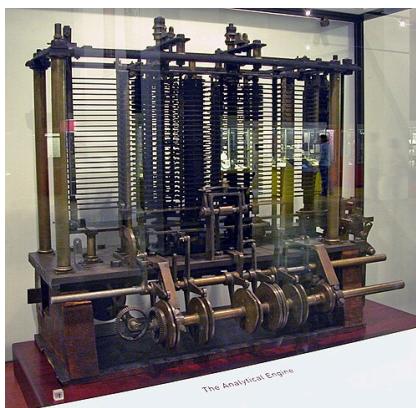


Figure 4: Métier à tisser de J.-M. Jacquard

Les concepts fondateurs

- Env. 1830 : Charles Babbage conçoit la machine analytique, un ordinateur programmable universel avant l'heure, inspiré du métier à tisser de Jacquard. Elle ne sera jamais construite faute de moyens techniques et financiers, mais ses plans étaient fonctionnels. Ada Lovelace y voit la possibilité d'écrire de véritables programmes (une suite d'instructions à mémoriser, exécuter et produire un résultat) et imagine déjà que la machine pourrait manipuler n'importe quel type de symbole — anticipant l'usage des ordinateurs pour le texte, la musique ou les arts: elle conçoit le premier algorithme en plus d'élargir la portée conceptuelle de la machine.



(a) Prototype (1871) non terminé de la machine analytique(a) « Programme » de calcul des nombres de Bernoulli dans une note d'A. Lovelace (1843)

- 1890 : Développement (par Herman Hollerith) des machines à cartes perforées pour le recensement américain, à l'origine d'IBM.

- 1936 : Alan Turing formalise, dans son article *On Computable Numbers*, la notion de *machine universelle*, un modèle théorique capable d'exécuter tout algorithme. Turing est ainsi une figure primordiale de l'histoire de l'informatique. Durant la Seconde Guerre mondiale, il contribue également au déchiffrement des codes secrets allemands (notamment *Enigma*)³.

Vers l'ordinateur moderne

- 1945 : John von Neumann formalise l'architecture du *programme enregistré*, où instructions et données sont stockées dans une même mémoire.

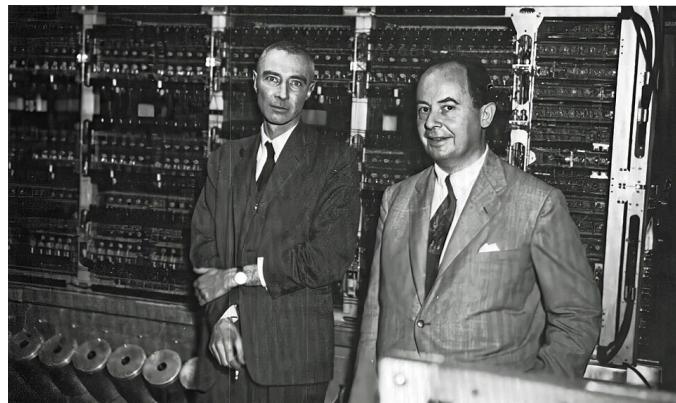


Figure 7: Oppenheimer et von Neumann à l'inauguration en 1952 de l'ordinateur construit pour l'*Institute for Advanced Study*.

- Années 1950 : IBM commercialise ses premiers ordinateurs ; FORTRAN⁴ (FORmula TRANslator) devient le premier langage de haut niveau utilisé massivement pour le calcul scientifique et l'ingénierie.

L'informatique et les sciences humaines

- 1949 : le jésuite Roberto Busa, avec IBM, lance le projet de l'*Index Thomisticus* (analyse assistée par ordinateur de l'œuvre de Thomas d'Aquin), considéré comme l'acte fondateur des humanités numériques.

Démocratisation et réseau

- Années 1960–1970 : naissance d'Internet, à la suite de projets militaires et universitaires (comme ARPANET, financé par le Department of Defense américain) qui mettent au point un réseau de communication décentralisé, capable de transmettre des paquets d'informations entre ordinateurs.

³C'est sur cet épisode que porte le film *The Imitation Game* (2014).

⁴Un bac à sable est disponible [ici](#).

- Années 1980 : diffusion des micro-ordinateurs personnels, qui ouvrent l'informatique à la recherche, l'enseignement et le grand public.

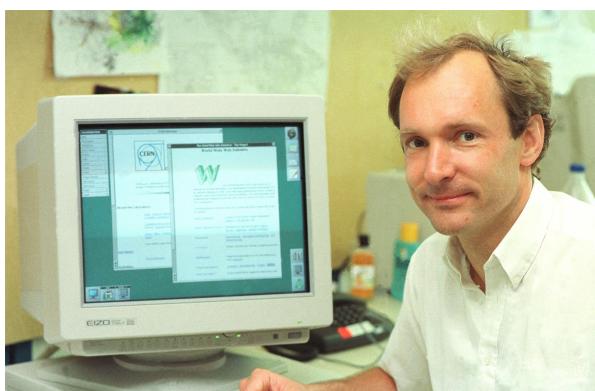


Figure 8: IBM PC 5150 avec clavier

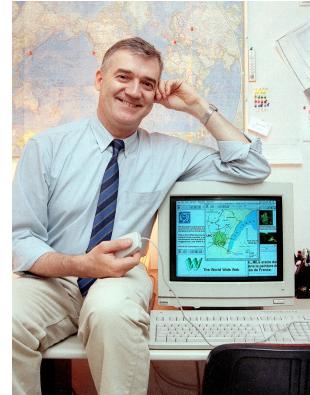
- 1990 : Tim Berners-Lee (avec la collaboration de Robert Cailliau) invente le World Wide Web au CERN, rendant possible la circulation mondiale des textes numériques.

i Note

Les concepts d'Internet et du Web sont complètement différents ! Le Web repose sur Internet, mais il en est une application.



(a) Tim Berners-Lee



(a) Robert Cailliau

- Années 1990-2000 : grands projets de bibliothèques numériques (Gutenberg, Perseus, Gallica) et standardisation de l'encodage de texte (TEI).

- 1998 : Google est créé par Larry Page et Sergey Brin. Il s'agit d'un moteur de recherche qui classe les pages Web selon leur popularité (PageRank).

Vers les humanités numériques contemporaines

- Années 2000-2010 : massification du numérique en SHS : OCR → HTR, bases textuelles, textométrie, fouille de texte ;
- Années 2010-2020 : montée en puissance de l'IA et des grands modèles de langage, offrant de nouvelles méthodes d'analyse, mais aussi de nouveaux enjeux critiques.

[Séance 2 : 09/09] COMPRENDRE son environnement

Retour sur la lecture

! Lectures obligatoires

- Sinatra & Vitali-Rosati ([2014](#))

Terminal, ligne de commande

La ligne de commande

- *Interfaces graphiques* = couche superposée à une interface de communication fondamentale : la commande ;
- Le *Terminal* que nous allons manipuler fait référence au logiciel avec lequel nous pouvons taper des lignes de commandes afin d'interagir avec le *système d'exploitation* ;
- *Shell* : interpréteur de commandes qui sert d'intermédiaire entre les utilisateur·ices et le système d'exploitation ;
 - Différents *shells* existent, le shell par défaut de la plupart des distributions Linux & MacOS X se nomme Bash.
- Les interfaces graphiques “traduisent” en lignes de commande les opérations que nous faisons (mais nous le cachent).
- Avantage : la ligne de commande est + rapide et + sûre.

OS : « logiciel faisant office d’interface [(graphique)] entre les composants informatiques et l’utilisateur, qui assure notamment l’exécution des programmes, l’allocation des ressources matérielles et le contrôle des opérations d’entrée-sortie sur les périphériques. » ([OQLF 2022](#)).

Exemple

Création d'un répertoire (= Dossier) sur le Bureau puis y déplacer un fichier que l'on vient de créer :

1. Clic droit, Dossier > Nouveau → une fenêtre apparaît ;
2. Saisir le nom du répertoire (**Test**) > cliquer sur **OK** ;
3. Déplacer le fichier **exemple.pdf** en le faisant glisser avec le bouton gauche de la souris sur l'icône du répertoire puis relâcher le bouton.

Création d'un répertoire (= Dossier) puis y déplacer un fichier que l'on vient de créer avec la ligne de commande :

1. ouvrir le terminal ;
2. `mkdir Test` → *Enter* ;
3. `mv exemple.pdf Test` → *Enter*.

La ligne de commande, plus rapide et plus sûre ?

- Interface graphique = surcouche logicielle → ralentit le système d'exploitation ;
- Interface graphique = logiciel complexe → *bugs* ;
- Interface graphique ne contient pas *tous* les équivalents qu'à la ligne de commande – certaines opérations ont été laissées de côté et ne pourront être effectuées par l'utilisateur·ice ;
- Séquence de lignes de commande, *e.g.*: `mkdir Test` ; `mv exemple.pdf Test` ;
- Présence de *wildcards* dans les lignes de commande, permettant de déclencher des opérations complexes sur un grand nombre de fichiers, *e.g.*: `mv *.pdf Test` déplacera tous les fichiers PDF dans le répertoire **Test**.

Testons

- Linux : chercher *terminal* dans les *outils système* (raccourci : **CTRL+ALT+t**) ;
- MacOS : chercher *terminal* dans les applications (toutes les commandes ne sont pas présentes) ;
- Windows : [installer Cygwin](#).

Où sont les fichiers sur un ordinateur ?

- Pourquoi c'est important ?
 - Interactions facilités avec les différents fichiers constituant un projet ;
 - Liens (relatifs et/ou absous) entre fichiers.

Arborescence

- Chaque système d'exploitation a son propre système d'organisation ;
- Il faut comprendre les systèmes de chemin ;
- Tout est arborescence

```
nak@Ubuntu-VM: $ tree -d ~/
/home/nak
├── Bureau
├── Documents
├── Images
├── Modules
├── Musique
├── Public
└── Videos
    ├── Firefox
    │   ├── 2008
    │   └── current --> 2009
    └── snap-desktop-Integration
        ├── Bureau
        ├── Documents
        ├── Images
        ├── Modules
        ├── Musique
        ├── Public
        └── Téléchargements
            ├── Videos
            └── 34
                ├── Bureau
                ├── Documents
                ├── Images
                ├── Musique
                ├── Public
                ├── Téléchargements
                └── Videos
                    ├── current --> 34
                    └── Téléchargements
                        └── Videos
34 directories
nak@Ubuntu-VM: $
```

Figure 11: Arborescence sur un système Ubuntu

- Votre disque dur = sorte de grande boîte où se trouvent d'autres boîtes (= répertoires/dossiers) ;
- *Racine* = boîte principale contenant toutes les autres ;
 - Mac, Linux : / ;
 - Widows : C:\.

L'emplacement du Dossier correspond à ce qu'on appelle son « chemin » (*path*).

Exemple d'un dossier « Vacances » :

- Unix : /Document Personnel/Images/Vacances
- Windows : C:\Dossier Personnel\Images\Vacances

Comprendre le fonctionnement global de l'arborescence permet de se repérer dans son ordinateur.

Nommer ses fichiers

- Quelques bonnes pratiques :
 - Les noms de fichier ne doivent pas contenir des caractères spéciaux, ni des accents, ni des espaces : l'ordinateur ne les reconnaît pas immédiatement, il doit les convertir, ce qui peut provoquer des erreurs ;
 - Les noms des fichiers doivent être significatifs ;
 - Les noms des fichiers ne doivent pas dépasser 80 caractères ;

- Faites attention aux extensions de fichiers ! Il est préférable de ne pas les masquer : information primordiale pour la machine ET les utilisateurs ;
- Exemple : 2025-09-09-hnu2000-seance02.md.

Chemins relatifs et chemins absous

Il est possible d'indiquer à la fois un chemin relatif, donc sans commencer par une barre oblique, ou en commençant par une barre oblique pour un chemin absolu.

Chemin absolu	Chemin relatif
/Users/Mathilde/Documents/Projet/index.html	./Documents/Projet/index.html
Emplacement complet à partir de la racine du système de fichiers	Chemin vers un fichier à partir du répertoire actuel (pwd)
Toujours valide, qu'importe le répertoire	Change selon où je me situe dans mes fichiers
Commence nécessairement par /	Ne commence pas par /

Astuces :

- / : on descend d'un niveau dans la hiérarchie des répertoires ;
- . . : on remonte vers un répertoire parent.

Les premières commandes

Quand je lance mon terminal, une invite de commande (*prompt*) apparaît,

e.g. `mathilde@crcen-mathilde-xps-13-7390:~:`

- l'utilisateur `mathilde` est connectée sur l'ordinateur `crcen-mathilde-xps-13-7390` ;
- le `~` indique qu'on se trouve dans le *home directory*.

Où suis-je ?

- pour savoir où l'on se trouve : `pwd`

 Note

`pwd` nous retourne un chemin absolu: le chemin indiqué commence en effet par une barre oblique.

Déplacer un fichier

- La commande `mv <source> <destination>` permet de renommer ou de déplacer un fichier (dans le terminal, c'est la même chose!);
- `mv mon-fichier.txt fichier.txt` renomme le fichier `mon-fichier.txt` en `fichier.txt` ;
- `mv /home/utilisateur/hnu2000/mon-fichier.txt /home/utilisateur/Documents/` déplace le fichier `mon-fichier.txt` dans le dossier `Documents` (il doit exister préalablement!).

Naviguer dans ses fichiers

- pour naviguer : `cd` suivi du chemin,
- exemple :
 - `cd Documents/cours/hnu2000` ;
 - `cd` : retour au *home directory*.

Note

Il est possible d'indiquer à la fois un chemin relatif, donc sans commencer par une barre oblique, ou en commençant par une barre oblique pour un chemin absolu.

Remarquez le changement de l'invite après l'exécution de la commande.

Lister des fichiers

- pour lister les fichiers : `ls` ;
 - ex. : `ls Test/> fichier.txt` ;
- pour lister les fichiers, dont les fichiers cachés : `ls -a`.

Créer un dossier

- `mkdir + le nom du dossier (MaKe DIRectory)`.

Créer un fichier

- `touch + le nom du fichier.`

Supprimer un fichier

- `rm mon-fichier.md` ;
- `rm -R mon-dossier` supprime le dossier et tout ce qu'il contient.

Afficher le contenu d'un fichier

- `cat mon-fichier.md`.

Comprendre une commande

- pour comprendre à quoi sert une commande : `man nom-de-la-commande` (ex. : `man rm`).

Quelques réflexes

- La touche TAB complète une commande ;
- La flèche du haut parcourt les dernières commandes utilisées ;
- CTRL + R pour rechercher une commande précédemment tapée ; la recherche se mettra à jour au fur et à mesure que vous tapez les premières lettres de la commande ;
- CTRL + L ou `clear` pour repartir sur un *nouveau* terminal.

Exercice

- créez un dossier HNU2000 (dans un dossier Cours par exemple) contenant un sous-dossier 02-exercice contenant lui-même un fichier `exemple.txt` ;
- naviguez dans le dossier 02-exercice ;
- revenez dans le dossier Cours ;
- déplacez le fichier `exemple.txt` dans le dossier HNU2000 ;
- renommez le fichier `exemple.txt` en `02_exercice1.txt` ;
- supprimez le dossier 02-exercice.

Quelques commandes avancées

- `history` : Pour afficher et parcourir l'historique des commandes que vous avez utilisées (seulement les commandes, pas les résultats de ces commandes) ;
- `less + fichier` : Affiche le contenu d'un fichier page à page (similaire à `cat`, plus lisible),
 - pour quitter : `q !`
- `cp <source> <destination>` copie un fichier ou un dossier :

- cp HNU2000/texte.txt Documents/divers/textes/texte.txt ;
 - cp -r textes Documents/divers ;
- wc permet de compter le nombre de lignes, de mots et de caractères dans un fichier texte ;
 - find permet de chercher dans les noms de fichiers et dossiers, et dans les fichiers eux-mêmes ;
 - && pour enchaîner les commandes.

Astuce

Testez la commande suivante : curl http://wttr.in/Montréal\?lang\=fr. Nous y reviendrons plus tard dans la session.

Bash

- Interpréteur en ligne de commande d'Unix (= Linux, Mac) ;
- Un *script* Bash (ou *shell*) est un fichier (en .sh) comportant plusieurs instructions/commandes ;
- Celles-ci seront exécutées les unes après les autres ;
- → Automatisation des tâches.

Exercices

Renommer tous ses fichiers automatiquement

Si vous désirez ajouter HNU ou vacances2025 devant chaque nom de fichier dans un dossier donné, créez un script :

- Créez (en ligne de commande), un fichier intitulé rename.sh ;
- Ajoutez-y le script suivant :

```
#!/bin/bash
# Ajoute "hnu_" devant chaque fichier

for file in *; do
    if [ -f "$file" ]; then
        mv "$file" "hnu_$file"
    fi
done
```

- Créez quelques fichiers de test (touch test1.txt test2.txt) ;
- Faites tourner le script : bash rename.sh.

Changer l'extension automatiquement

Le script :

```
#!/bin/bash
# Change l'extension .md en .txt pour tous les fichiers

for file in *.md; do
    if [ -f "$file" ]; then
        mv "$file" "${file%.md}.txt"
    fi
done
```

Générer du texte

Testez plusieurs scripts Bash. Pour ce faire, créez des fichiers nom-du-fichier.sh, ajoutez-y le contenu indiqué ci-dessous, et faites les tourner en tapant, dans votre terminal, bash nom-du-fichier.sh :

```
echo "texte";

valeur="texte";
echo ${valeur}

valeur="texte";
echo "je suis du ${valeur}"
```

Jouer avec les commandes de base

- Afficher la date :

```
horaire=`date`
echo "Jounal de bord du $horaire"
```

- Afficher la situation :

```
lieu=`pwd`
echo "Je me trouve toujours à bord du $lieu"
```

- Demander mon identité : (hors interface en ligne)

```
identite=`whoami`  
echo "Je me prénomme $identite"
```

- Aligner du texte :

```
T[0]='texte1'  
T[1]='texte2'  
echo ${T[0]} ${T[1]}
```

- Création d'une interaction :

```
echo "To be or not to be ?"  
      select i in être non-être; do  
        if [ "$i" = "être" ]; then  
          echo "Sois";  
          break  
        elif [ "$i" = "non-être" ]; then  
          echo "Tant pis"  
          break  
        else  
          echo "Pas de troisième choix dans ce script"  
        fi  
      done
```



Astuce

D'autres exemples [ici](#).

WYSIWYM >< WYSIWYG ?

Deux grandes familles de systèmes de saisie de documents textuels :

WYSIWYM	WYSIWYG
<i>What You See Is What You Mean</i> éditeur de texte qui présente le texte et ses informations de mise en forme sans les interpréter	<i>What You See Is What You Get</i> éditeur de texte qui présente un texte enrichi avec sa mise en forme, de manière simultanée

Le logiciel de traitement de texte *Word* est-il WYSIWYM ou WYSIWYG ?

WYSIWYM

- Moindre confusion entre mise en forme et indication de la structure logique ;
- La mise en forme apparaît après *parsing* du document (création d'un PDF ou ouverture dans un navigateur) ;
- Le paramétrage de la mise en forme peut être géré dans un fichier externe (ex: CSS pour HTML).



Note

Si nous avons le temps, lecture en classe de [ce billet](#).

Mettre en forme =/= structurer

Mise en forme	Structure logique
Du texte en gras	Un titre
Du texte en italique	Un titre, un mot étranger
Du texte entre guillements	Une citation

VSCodium : installation et manipulations

C'est quoi Codium ?

- Version open source de Visual Studio Code (VS Code)

VSCodium is a community-driven, freely-licensed binary distribution of Microsoft's editor VS Code.

- VS Code : éditeur de code développé par Microsoft. Gratuit, léger et puissant. Largement utilisé par la communauté des développeur·ses pour faire de la programmation.

Extensions

Suivre [ce tutoriel](#) (pour installer French Language Pack for VS Code, French – Code Spell Checker et LTeX, à configurer en français).

[Séance 3 : 16/09] CHERCHER: corpus et ressources numériques

Retour sur les lectures et écoutes

! Lectures obligatoires

- France Inter ([s. d.](#))
- Huma-Num ([2021](#))

Une histoire d'internet et du Web

Nous avons évoqué, lors de la première séance, Internet et le Web. Qu'en avez-vous retenu ? S'agit-il de la même chose ?

Ce ne sont pas tant les « nouvelles technologies » en général, mais le réseau en lui-même qui a bouleversé notre rapport à la connaissance. ([Mille, 2014](#))

Origines

- 1962-1968 : ARPAnet
- 1969-1978 : Internet
- années 1970 et 1980 : accès aux machines
- 1984 : un réseau fonctionnel

Plus de détails dans ([Mille, 2014](#)).

Principes : TCP/IP

TCP/IP :

- TCP (*Transmission Control Protocol*) ;
- IP (*Internet Protocol*).

Une pile de protocoles :

- **physique** : les câbles (cuivre ou fibre), éventuellement radio ;
- **liaison** : Ethernet ou Wireless Ethernet (Wifi), il s'agit de déterminer comment les paquets sont acheminés ;
- **réseau** : c'est la partie IP qui permet d'acheminer des paquets en donnant des adresses à toutes les machines connectées sur un réseau ;
- **transport** : c'est la partie TCP, pour transférer les informations découpées en paquets et reconstituées en vérifiant qu'il ne manque rien ;
- **application** : c'est là qu'apparaissent les services qu'on connaît : le Web avec HTTP, les emails avec SMTP ou IMAP, ou encore le FTP pour partager des fichiers.

Astuce

Pour en savoir plus sur Internet et ses protocoles, voici une série de vidéos très complètes : <https://iletaitunefoisinternet.fr/>.

Expérimentation

Découvrir la *route* pour accéder à un serveur :

- utiliser `tracepath` (ou `traceroute`) en ligne de commande
- lancer la commande `tracepath umontreal.ca` ou `traceroute umontreal.ca`
- analyser les résultats

Le Web

- Internet ≠ Web
- le Web = une application d'Internet
- un protocole (HTTP) et des langages (HTML/CSS/JavaScript)

Le Web est une application d'Internet permettant de publier et de consulter facilement des informations.

Note

Le CERN a restauré la première page du Web.

Exercice

- comment lire une page web via un autre outil qu'un navigateur ?
- utiliser la commande curl dans le terminal ;
- exemple 1 : `curl https://mathildevrst.github.io/HNU2000-A25/Plan-Cours.html` ;
- exemple 2 : `curl https://theread.me/raw-permalinks-for-accessibility/`.

Les navigateurs et moteurs de recherche

Au principe d'autorité qui a fait la force du PageRank, Google substitue de plus en plus un principe d'efficacité qui renvoie de manière toujours plus appropriée vers l'internaute les choix que l'algorithme a appris de ses comportements. ([Cardon, 2013](#))

- Navigateur ≠ moteur de recherche⁵ !
 - Navigateur : application, installée sur l'ordinateur, qui permet d'afficher les informations contenues sur les sites Web ;
 - Moteur de recherche : application qui permet de localiser des informations sur le Web, [...] outil devenu indispensable à partir du moment où les ressources Web, devenues trop nombreuses, ont rendu difficile l'indexation des contenus en ligne.

Quels navigateurs utilisez-vous ? Pourquoi ?

Le Web, au commencement étaient les annuaires

- Dans les années 1990, les premiers répertoires (Yahoo!, DMOZ) proposaient une liste classée de sites web sous forme d'une arborescence thématique ;
- La démarche est différente d'aujourd'hui : on naviguait dans une hiérarchie, on ne formulait pas encore une requête.
- Avantages : navigation linéaire par thématiques, parcours dans une arborescence logique, aperçu potentiellement exhaustif de l'existant.
- Inconvénients : parfois une seule entrée pour un résultat qui concerne plusieurs thématiques, recherche fastidieuse ; explosion du Web rendant le modèle vite obsolète.

⁵Les deux définitions ci-dessous sont issues de ([Debouy, 2025, p. 125](#)).



Figure 12: Yahoo! en 2001

Explorer, indexer, classer

Un moteur de recherche recense et référence des pages Internet : pour cela, il utilise un logiciel, un robot d'indexation (dits *crawlers* ou *spiders*) qui parcourt l'ensemble du Web :

1. Exploration : exploration régulière du Web par le *crawler* qui suit tous les liens trouvés puis analyse les pages jugées intéressantes ;
2. Indexation : extraction de mots-clés des pages visitées et conservation d'une copie (= *cache*) de ces pages ;
3. Classement : classement, par le moteur de recherche, des résultats en fonction de leur pertinence. C'est là qu'interviennent les algorithmes, qui tiennent compte par exemple de la popularité d'une page (combien d'autres sites font un lien vers elle), mais aussi de centaines d'autres critères que Google garde secrets.

Sundar Pichai, président-directeur général de Google, tente d'expliquer au Congrès Américain comment fonctionne Google

<https://www.youtube.com/watch?v=8fP95zyPePU>

Quelques moteurs de recherche

Google

- Le plus utilisé, merci le *PageRank*, algorithme mathématique utilisé pour déterminer la pertinence des pages Web ;
- Attention aux premiers liens qui peuvent être sponsorisés ;

- Personnalisation des résultats : deux utilisateurs effectuant la même requête n'obtiendront pas les mêmes résultats (prise en compte des recherches et activités passées, ainsi que du profil des utilisateurs (situation géographique, âge, langue, etc) ([Ochigame & Ye, 2021](#))) ;
- Quid des données personnelles ?

Alternatives

Quelques moteurs de recherche n'enregistrent pas les recherches effectuées et n'exploitent pas les données personnelles à des fins publicitaires ; ne filtrent pas les résultats de recherche en fonction des profils des utilisateurs → neutralité des recherches :

<i>Qwant</i>	<i>DuckDuckGo</i>	<i>Startpage</i>
“Le moteur de recherche dont vous êtes l’utilisateur, pas le produit”	“Le moteur de recherche qui ne vous espionne pas”	“Le moteur de recherche le plus privé au monde”
Exploite simplement les banques de données réalisées par les <i>crawlers</i>	Métamoteur qui interroge les banques de données des autres moteurs de recherche	← Idem
Français & Vie privée	Gain de temps & Protection des données	← Idem

Google Scholar

- Moteur de recherche généraliste ≠ Moteur de recherche d’articles scientifiques ;
- Généralement, accès au texte intégral ;
- Mais, pas de possibilité de tri/filtre des résultats ;
- Pas de vérification du caractère scientifique des articles ;

Isidore

- Projet de recherche, 2011 < TGIR Huma-Num ;
- objectifs : mettre à disposition un outil de recherche adapté aux sciences humaines ;
- ambition : proposer des fonctionnalités pour faciliter la recherche documentaire ;
- S’appuie sur le principe du Web de données ;
- Se distingue des autres moteurs de recherche :
 - Moissonage ciblé et des métadonnées et données scientifiques structurées selon des standards internationaux et disponibles en accès libre
 - Indexation des données non structurées et structurées ;
 - Normalisation des métadonnées et enrichissement des données en s’appuyant sur des référentiels reconnus ;

- Mise en évidence des sources de données indexées ;
- Récupération des actualités ;
- Algorithmes basés sur des référentiels selon des techniques documentaires \neq PageRank.

Exercice

- constituez une bibliographie dans Isidore.science sur les humanités numériques ;
- pour cela vous devez vous créer un compte Isidore (et donc HumanID) ;
- utilisez les fonctionnalités de la recherche avancée ;
- utilisez les fonctions de tris et de facettes ;
- partagez avec moi votre « bibliothèque ».

Discussion : Que manque-t-il à Isidore ? Quels sont ses défauts ?

Google et le PageRank

En 1998, Google introduit le PageRank, qui attribue une importance à une page selon le nombre et la qualité des liens qui pointent vers elle (= système de mesure quantitative de popularité d'une page web). C'est inspiré par la mesure des articles académiques (Science Citation Index).

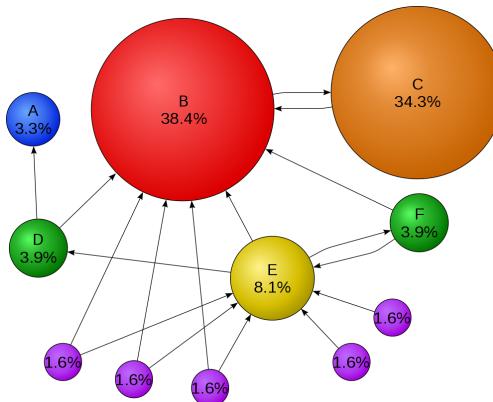


Figure 13: Schéma du PageRank

Cela fonde un principe d'autorité : les pages citées par beaucoup d'autres deviennent les plus visibles. Progressivement, Google combine ce principe avec d'autres critères (localisation, personnalisation).

Les évolutions des moteurs de recherche

Il y a plusieurs évolutions majeures des moteurs de recherche ces dernières années :

- Interface utilisateur (UI) : simplification radicale de la page Google, disparition progressive des options avancées ;
- Aide à la formulation : autocomplétion, suggestions liées aux recherches fréquentes ;
- Web sémantique et Knowledge Graph (2012–) : affichage de données structurées (fiche Wikipédia, horaires, météo,...) directement dans les résultats ;
- Personnalisation : traçage des comportements, adaptation aux historiques et aux contextes ;
- SEO (Search Engine Optimization) : pratiques d'optimisation par les sites pour “plaire” à Google ;
- IA générative (2023–) : moteurs qui ne se contentent plus de lister des pages, mais produisent directement des résumés de réponse (Google SGE, Perplexity).

Phénomène de simplification :

- Suppression des options → interface minimale, recherche “intuitive” ;
- Autocomplétion → orientation subtile des requêtes ;
- Pertinence des premiers résultats → effet d'écrasement : très peu d'utilisateurs consultent la deuxième page ;
- Affichage de données liées (web sémantique) → l'utilisateur n'a plus besoin de cliquer sur les sites sources ;
- IA générative → bascule majeure : le moteur devient assistant ou agent conversationnel, qui filtre et reformule le Web au lieu de seulement l'indexer ;
- Tendance globale → l'information va de plus en plus vers l'utilisateur, au risque de réduire sa diversité d'accès.



Figure 14: Outil de réservation SNCF, capture d'écran, 18 août 2025

Vade-Mecum d'une recherche efficace

1. Commencer simple
 - Tapez quelques mots-clés principaux de votre sujet ;
 - Utilisez des mots précis plutôt que des phrases longues ;
 - Google ignore généralement les mots très fréquents (le, la, de...).
2. Utiliser les guillemets pour les expressions exactes
 - "expression exacte" → Google recherche ces mots dans cet ordre précis ;
 - e.g.: "humanités numériques" ne retournera que les pages contenant exactement cette expression.

3. Limiter la recherche à un site ou un domaine

- `site:lesite.com` requête → Recherche uniquement sur ce site ;
- e.g.:`site:umontreal.ca "humanités numériques"` → pages sur l'Université de Montréal concernant les humanités numériques.

4. Chercher un type de fichier spécifique

- `filetype:pdf` requête → Cherche des fichiers PDF uniquement ;
- e.g.:`filetype:pdf "édition numérique"` → documents PDF sur l'édition numérique.

5. Chercher dans le titre d'une page

- `intitle:mot` → Page dont le titre contient ce mot ;
- e.g.:`intitle:"humanités numériques"` → pages dont le titre contient l'expression exacte.

6. Les opérateurs

- AND⁶, OR, NOT permettent de préciser la recherche. Des parenthèses servent à grouper des termes ou des opérateurs ;
 - e.g. : `"humanités numériques" AND "édition" -cours` → pages sur l'édition dans les humanités numériques mais sans cours ;
 - e.g. : `("humanités numériques" OR "digital humanities") AND "édition"` → pages qui contiennent soit "humanités numériques", soit "digital humanities", mais qui contiennent aussi "édition".
- ? : Remplace un caractère ;
 - e.g. : `econom?` recherche `économie`, `economic`, `economia`.
- * : Élargit la recherche aux variantes d'un même mot ;
 - e.g. : `pers*` recherche `personnage`, `personnalité`, `persona`,...
- > < = : précise une période chronologique.

Astuce

Les formulaire de recherche avancée des bibliothèques utilisent ces opérateurs!

⁶Si les mots-clés ne sont pas reliés par un opérateur, la plupart des outils de recherche considèrent que les termes sont reliés par AND.

Exercice

Dans votre moteur de recherche favori :

- Cherchez une expression exacte ;
- Cherchez là uniquement sur un site spécifique ;
- Cherchez un type de fichier particulier ;
- Cherchez des pages dont le titre contient un mot-clé ;
- Combinez plusieurs opérateurs pour affiner la recherche ;
- Cherchez toutes les pages qui parlent d'humanités numériques et d'édition sur le site web de l'Université de Montréal.

Une alternative aux moteurs de recherche : les flux RSS

- *Really Simple Syndication* ;
- Ressources dont le contenu est mis à jour de manière automatique selon les modifications d'un site Web ;
- On s'*abonne* à un flux en l'ajoutant dans un *agrégateur* qui vérifiera automatiquement si de nouveaux contenus ont été mis en ligne ;
 - Parmi les agrégateurs, des applications Web (Netvibes, Feedly, Inoreader, FreshRSS,...) ou des extensions de navigateur (Feedbro,...) ;
- Avantage : on suit les actualités d'un site sans devoir s'y redire manuellement.

[Séance 4 : 23/09] ORGANISER & BIBLIOGRAPHIER: métadonnées et formats

Retour sur les lectures

! Lectures obligatoires

- Collectif ([2018](#))

Les métadonnées

- Données sur des données ;
- Sont des métadonnées les notices bibliographiques, le format d'un fichier, la date, la géolocalisation d'une photo,...

Composé à l'aide du préfixe grec *meta*, le terme désigne à proprement parler les données à propos des données : elles servent à référencer, identifier et partager un document. Elles indiquent notamment comment, quand, où et par qui les données ont été recueillies, mentionnent leur disponibilité et leur mode de distribution, leur taille, etc. ([Debouy, 2025, p. 160](#))

Pourquoi les métadonnées ?

- Trouver : elles rendent les données repérables (moteurs de recherche, catalogues) ;
- Comprendre : elles donnent du contexte (qui a produit la donnée, quand, dans quelles conditions) ;
- Réutiliser : elles précisent les droits, les formats, les relations avec d'autres données ;
- Préserver : elles facilitent l'archivage et l'interopérabilité à long terme.

Les différents formats textuels

Les formats : un peu d'étymologie

- Dimensions d'une feuille de papier, d'un feuillet ;

- Du latin *formare* : donner une forme, former, conformer ; arranger, organiser, régler ; façonner ;
- Extension à de nombreux domaines : imprimerie, photographie, cinéma, **informatique**.

Vers une définition du format

- Ce qui donne une forme et un ensemble de caractéristiques à un ensemble d'informations (image, écriture, son, etc.) ;
- La dimension (format A4 ou *letter* par ex.) qui va impliquer un agencement ;
- Une manière de représenter des données (pdf, docx) qui impose une structure ;
- **interopérabilité**.

Structure définie de données contenues dans un fichier, établie selon des règles qui régissent notamment leur sauvegarde, leur affichage et leur manipulation par un logiciel compatible. ([OQLF, 2022](#)).

En informatique, au niveau le plus fondamental, tout est exprimé dans un alphabet numérique binaire : 0 et 1. Un format, c'est une convention qui établit la correspondance entre une certaine succession de 0 et de 1, et quelque chose d'autre : par exemple une couleur, ou une lettre de l'alphabet, ou la position d'un pixel sur un écran. L'expression « format texte » désigne une catégorie de formats pour lesquels le contenu en binaire des fichiers encode des caractères textuels uniquement. ([Arthur Perret](#))

Format et édition savante

- Production de documents scientifiques qui ont des caractéristiques exigées par les normes de l'édition savante :
 - appareil critique accompagnant le texte (notes de bas de page, bibliographie, index, etc.) ;
 - citabilité (métadonnées → identification) ;
 - structuration (normalement) rigoureuse (balisage sémantique, non ambigu) ;
 - pérennité ;
- Les formats utilisés doivent répondre à ces exigences, permettant notamment d'expliciter la fonction de chaque élément du document (titre, citation, métadonnées, références bibliographiques,...)

Format et extension

À chaque format de fichier est associée une extension de nom de fichier qui permet de l'identifier. [Cette] extension de nom de fichier indique au système d'exploitation quel logiciel d'application peut ouvrir le fichier et quelle icône afficher. ([OQLF, 2022](#))

L'extension d'un fichier texte indique à l'utilisateur et à l'ordinateur quel est son format, c'est-à-dire :

- Comment sont organisées les informations relatives au fichiers et les métadonnées ;
- Avec quel logiciel l'ouvrir ;
- S'il contient ou non des informations de mise en page qu'il faut interpréter.

Formats libres, formats propriétaires

Formats libres	Formats propriétaires
ouverts	fermés (parfois ouverts)
spécifications techniques publiques	spécifications techniques contrôlées par une entité privée
pas de restriction d'accès	usage limité
définis par les pratiques	conditionnés par les pratiques
exigences de recherche	exigences commerciales
.txt ; .odt ; .tex ; .html ; .svg ; .png ; .epub ; .zip	.pdf ; .docx ; .pptx ; .xlsx ; .ai ; .jpeg ; .psd ; .mp3

Le format n'est pas neutre !

Le format est politique en ce qu'il porte un modèle de l'information.

Zoom sur le libre

- Discussion : Qu'en savez-vous ? Intérêts, avantages, inconvénients,... ?
- Quid des productions scientifiques des chercheurs publiées dans des revues scientifiques en limitant l'accès ?
- Question du partage et de la diffusion des connaissances scientifiques ;
- Question de la possibilité du lien entre les données

En agrégeant des contenus, en liant les publications avec les données de la recherche et à bien d'autres objets, les technologies du Web sémantique permettent la construction de nouveaux savoirs. ([Debouy, 2025, p. 153](#))

Le cas des revues

- Depuis les années 2000, développement des revues en accès ouvert : plus de 10.000 à ce jour !
 - ex. : [OpenEdition](#) : portail (français) de ressources électroniques en SHS ;
 - ou [HAL](#) : archive ouverte pluridisciplinaire.

Différentes voies possibles :

1. *green open access* :

- autoarchivage permettant l'accès libre et gratuit aux articles scientifiques ;
- si l'article a déjà été publié par un éditeur :
 - celui-ci le libère
 - mais, souvent, période d'embargo afin d'inciter les utilisateurs à continuer à souscrire des abonnements

2. *gold open access* :

- l'auteur (ou son institution d'appartenance) paie pour sa publication ;
 - // auteur-payeur

3. les *épi-revues* :

- Voie dorée par l'hébergement de revues en *open access*, et
- Voie verte via le processus de soumission des articles qui se fait sur une archive ouverte ;
- Pas de frais d'accès pour les lecteurs ni de frais de publication pour les auteurs ;
- Les articles sont immédiatement consultables en accès ouverts (avant même l'évaluation par les pairs) ;
- Finalement, validation de l'article pour publication dans la revue : nouvelle version soumise sur l'archive + label de la revue ;
- Ex. : <https://open-research-europe.ec.europa.eu/articles/5-16/v1>.

Qu'est-ce qu'une bibliographie ?

L'importance de structurer ses références bibliographiques

- Communauté scientifique & standards : inscrire votre travail dans les standards scientifiques d'une communauté ;
- Éthique : qui réfère bien respecte le travail cité ;
- Pérennité : pour retrouver vos sources – médiation d'un travail de sources à la base de votre recherche ;
- Esthétique : une bibliographie imparfaite fait tâche dans un travail de recherche ;
- Personnel : vous avez le choix de structurer ces références selon le type de style bibliographique.

Style de citation vs style bibliographique

Style de citation	Style bibliographique
règles utilisées pour citer des sources dans le texte	présentation complète des sources dans une bibliographie ou une liste de références
APA : (Dupont, 2020, p. 45)	APA : Dupont, J. (2020). Titre du livre. Éditeur.
Vancouver : (1)	Vancouver : Dupont J. Titre du livre. Lieu de publication: Éditeur; 2020. 1000p.

Répertoire des styles : <https://www.zotero.org/styles>

Exemple

APA 7th - anglophone	Merzeau, L. (2013). Éditorialisation collaborative d'un événement. <i>Communication et organisation. Revue scientifique francophone en Communication organisationnelle</i> , 43, Article 43.
APA 7th - francophone	Merzeau, L. (2013). Éditorialisation collaborative d'un événement. <i>Communication et organisation. Revue scientifique francophone en Communication organisationnelle</i> , (43), 105-22.
Modern Language Association 9th edition	Merzeau, Louise. "Éditorialisation collaborative d'un événement." <i>Communication et organisation. Revue scientifique francophone en Communication organisationnelle</i> , no 43, June 2013, pp. 105-22.
Chicago Manual of Style 17th edition	Merzeau, Louise. 2013. "Éditorialisation collaborative d'un événement." <i>Communication et organisation. Revue scientifique francophone en Communication organisationnelle</i> , no 43 (June): 105-22.

Ensemble de données !

Comment créer une bibliographie ?

- Plusieurs milliers de styles bibliographiques : <https://www.zotero.org/styles> (choisir la *norme* que l'on va utiliser) ;
- Exemple : « Université de Montréal - APA (Français – Canada) » ;
 - Pour modifier le style par défaut de Zotero : Édition > Préférences > Citer ;
- Création d'une bibliographie : chaque référence ou groupe de références peut être récupéré sous différents formats pour être intégré dans un document via un traitement de texte par exemple.

Plus d'informations : https://www.zotero.org/support/fr/creating_bibliographies

Zotero

Qu'est-ce que Zotero ?

- Un logiciel et un service qui permet de gérer des références bibliographiques : conserver, organiser, citer et créer des bibliographies ;
- Un logiciel libre, et gratuit ;

- Un outil qui peut être collaboratif ;
- Un logiciel + une extension de navigateur pour récupérer des références.

Fonctionnalités

- Collecter : collecter des références bibliographiques en ligne et les conserver ;
- Organiser : organiser ses références en collections et sous-collections, et en attribuant des tags et des marqueurs ;
- Citer : citer ses références facilement et les exporter/intégrer dynamiquement dans des documents ;
- Synchroniser : synchroniser ses références avec un compte en ligne ;
- Collaborer : échanger ses références avec d'autres utilisateurs.

Un logiciel libre

Zotero est logiciel libre qui peut être téléchargé puis installé sur tous les ordinateurs.

Il faut également utiliser un connecteur pour récupérer des références sur le web puis faire le lien avec le logiciel qui les conserve.

Utilisation générale

- Récupérer des références depuis un navigateur web (grâce à l'extension Zotero) ;
- Une icône apparaît pour enregistrer une référence (plusieurs types de documents) ;
- La référence est enregistrée dans le logiciel Zotero sous la forme d'une notice ;
- Il est possible de compléter la notice s'il manque des informations ;
- La synchronisation permet de sauvegarder les données en ligne (pour cela il faut créer un compte Zotero).

Fonctionnalités proposées avec un compte

- Synchronisation : vos références sont sauvegardées et disponibles en se connectant depuis n'importe quel ordinateur ;
- Sauvegarde ;
- Interface web : un peu moins pratique que le logiciel ;
- Afficher ses références ;
- Créer ou participer à des groupes ;
- Etc.

À quoi sert un groupe ?

Un groupe est une collection à laquelle plusieurs personnes peuvent participer :

- Partager des références ;
- Ouvrir les contributions ;
- Afficher les contributions ;
- Plusieurs rôles dans un même groupe : consulter, créer, modifier, administrer ;
- Le groupe peut être public avec des permissions plus ou moins ouvertes

Installation

- Installer le logiciel Zotero en suivant les recommandations du site web officiel ;
- Installer l'extension pour Firefox/Chrome/Safari ;
- Créer un compte en ligne ;
- Connecter votre logiciel à ce compte (Édition > Préférence > Synchronisation).

De nombreuses bases de données compatibles

De nombreux catalogues, bases de données, revues en ligne, journaux, blog, réseaux sociaux proposent des références :

- Sudoc ;
- Gallica ;
- OpenEdition, revues.org, Hypothèses ;
- Érudit ;
- Google Livres, Google Scholar, YouTube ;
- Internet Archive ;
- Cairn.info ;
- etc.

Description d'une utilisation classique

Prérequis : logiciel installé et extension installée (par exemple dans Firefox).

- Ouvrir le logiciel (Zotero Standalone) ;
- Rechercher un document sur le web ;
- Enregistrer la référence via l'extension du navigateur ;
- Vérifier dans le logiciel si la référence convient (et si elle est placée au bon endroit) ;
- Compléter les informations.

Quelques points importants

- Pas besoin d'être en ligne pour utiliser Zotero (mais vous ne pourrez pas afficher de pages web ou synchroniser vos collections) ;
- Les fonctions de Zotero sont nombreuses, n'hésitez pas à les tester et à les utiliser !
- Zotero permet de conserver et de gérer des références de pages web, mais ce n'est pas sa fonction première ;
- N'oubliez pas d'ouvrir le logiciel Zotero pour enregistrer vos références.

Collector

- Via la navigateur en cliquant sur l'icône : Zotero récupère les métadonnées de la page web (page web qui présente un document) ;
- Si le texte intégral est disponible en PDF, Zotero le récupère et le sauvegarde ;
- Si la page web contient plusieurs références (par exemple une page de résultats), alors l'icône est un dossier et il est possible de sélectionner les résultats que vous souhaitez conserver ;
- Zotero peut ajouter une référence automatiquement grâce à un identifiant – ISBN ou DOI : cliquez sur « Ajouter un élément par son identifiant » dans la barre d'outils Zotero, puis tapez le numéro d'identification, l'élément sera ajouté à votre bibliothèque.

Créer

Vous pouvez aussi créer manuellement des références en sélectionnant le type de document que vous souhaitez référencer et en remplaçant les champs de la notice.

Organiser

Collections :

- Dans colonne de gauche, « Mes collections » ou « Ma bibliothèque », il est possible de créer plusieurs collections ;
- Créez une nouvelle collection en cliquant sur le bouton « Nouvelle collection » ;
- Il est possible de créer, renommer ou supprimer une collection (via un clic droit sur une collection) ;
- Pour organiser vos références il suffit de les déplacer d'un dossier d'une collection à une autre ;
- Les collections peuvent comprendre des sous-collections !

Tags et marqueurs :

- Les tags et les marqueurs servent à ajouter des informations de description ;
- Vous pouvez vous créer votre propre système de marqueurs et de tags pour organiser plus finement vos références ;

- Certaines références importées comportent parfois des marqueurs, vous pouvez vous en inspirer ou les supprimer ;
- Une recherche dans Zotero interroge tous les champs, y compris les marqueurs et les tags !

Champ « note » :

- Il permet de saisir du texte libre lié à une référence précise ;
- Il peut y avoir plusieurs notes pour une même référence ;
- Il est possible de les visualiser rapidement, de les supprimer, et elles sont synchronisées si vous avez un compte Zotero ;
- Une recherche dans Zotero interroge tous les champs, y compris les notes !

Exercice : Zotero

Exercice 1

- Créer une référence depuis Zotero : un livre de votre choix (titre, auteur·trice, maison d'édition, lieu, date, langue, ISBN) ;
- Intégrer 4 références : un article, un livre, un chapitre et une thèse ;
- Déplacer toutes vos références dans une sous-collection “HNU2000”.

Exercice 2 : Traitement de texte

- Citations et bibliographies dynamiques dans des outils de traitement de texte (Word ou LibreOffice) :
 - Ouvrez LibreOffice/Word et vérifiez que vous disposez bien des options de Zotero (dans la partie supérieure gauche) ;
 - si vous n'avez pas l'extension Zotero pour LibreOffice, installez là via le logiciel Zotero : Édition > Préférences > Citer > Traitements de texte > Installer le module LibreOffice
 - vous pouvez ajouter une citation d'une référence préalablement enregistrée dans Zotero ;
 - une fois des citations ajoutées, vous pouvez créer une bibliographie de ces citations ;
 - si vous modifier vos références dans Zotero, vous pouvez mettre à jour votre document LibreOffice.

BibTeX

BibTeX est un format de fichier structurant une bibliographie :

- Format qui a été créé en 1985 pour gérer les bases de données bibliographiques dans des fichiers LaTeX ;
- Le format BibTeX fonctionne par entrée bibliographiques : chaque entrée correspond à une description d'un document avec des items « mot-clef = valeur ».

Fichier `ref.bib` contenant autant d'entrées que de références :

```
@article{merzeauEditorialisationCollaborativeDun2013a,
  title = {Éditorialisation collaborative d'un événement : l'exemple des Entretiens du},
  volume = {43},
  issn = {1168-5549},
  shorttitle = {Éditorialisation collaborative d'un événement},
  url = {https://shs.cairn.info/revue-communication-et-organisation-2013-1-page-105},
  doi = {10.4000/communicationorganisation.4158},
  language = {fr},
  number = {1},
  urldate = {2024-10-10},
  journal = {Communication \& Organisation},
  author = {Merzeau, Louise},
  year = {2013},
  note = {Publisher: Presses universitaires de Bordeaux}
  Section: Info, Communication},
  pages = {105--122},
}
```

Better BibTeX

Better BibTex (<https://retorque.re/zotero-better-bibtex/installation/>) ajoute à Zotero des fonctionnalités très pratiques :

- Affichage des clés de citation (identifiants BibTeX uniques et stables) ;
- Nouveaux formats d'export ;
- **Exports dynamiques** de bibliographies (synchronisation automatique des fichiers `.bib`).

Installez Better BibTeX

<https://retorque.re/zotero-better-bibtex/installation/>

Exercices

- Créez une collection avec plusieurs références que vous exportez au format BetterBibTeX avec l'option « Garder à jour » ;
- Ajouter une référence à cette collection ;
- Est-ce que le fichier BibTeX a été modifié ? ;
- Verrouillez une clé de citation d'une référence de ce export.

[Séance 5 : 07/10] REPRÉSENTER le texte

(I): formats légers

Retour sur les lectures

! Lectures obligatoires

- Vitali Rosati et al. ([2020](#))

Nous le ferons dans la deuxième partie du cours.

Objectifs de la séance

Théorie :

- Comment représenter, sous forme numérique, un texte ?
- Les avantages des formats légers ?
- Principes du balisage.

Pratique :

- Tuto Markdown ;
- Tuto Stylo ;
- Bonus : un peu de HTML.

Représenter le texte ?

- Représenter >< Montrer ?
- Le texte et son support !

Ce poème est-il le même ?

1	Catullus
2	Ad Lesbiam (V)
3	
4	-----
5	
6	Vivamus mea Lesbia, atque amemus,
7	rumoresque senum seueriorum
8	omnes unius aestimemus assis!
9	soles occidere et redire possunt:
10	(5) nobis cum semel occidit breuis lux,
11	nox est perpetua una dormienda.
12	

* ⌂ ⌂ 0 ⌂ 0 Quarto: 1.5.57 UTF-8 LF Texte brut

Catullus

Ad Lesbiam (V)

Vivamus mea Lesbia, atque amemus,
 rumoresque senum seueriorum
 omnes unius aestimemus assis!
 soles occidere et redire possunt:
 (5) nobis cum semel occidit breuis lux,
 nox est perpetua una dormienda.

Le texte et l'ordinateur

- Une suite de symboles ;
- Qui peuvent être structurés ;
 - À nous de définir cette structure !
- Dont il ne comprend pas le sens.

Comment choisir un format, une syntaxe, qui rende justice à un texte donné ?

Encodage, format, balisage

- **Encodage** : façon dont les caractères sont traduits en nombres (binaires), où chaque *encodage* correspond à une *norme* (ASCII, UTF-8, etc). [Plus d'informations ici](#) ;
- **Format** : organisation du contenu à l'intérieur d'un fichier (cf. séance précédente) ;
- **Langage de balisage** : façon dont le texte lui-même est structuré par des signes visibles.

Chaque format et langage impose une manière de penser le texte : .docx pour la mise en page (et encore...), .html pour la hiérarchie entre les éléments, .md pour la lisibilité et interopérabilité.

Une courte histoire du balisage

```
<chapter><title>Introduction to SGML</title>
<section><title>The SGML Declaration</title>
<subsection>
```

SGML, HTML et XML

- **SGML** : décrire avec des balises ;
- **HTML** : structurer pour afficher ;
- **XML** : l'arbre sémantique.

L'histoire des langages de balisage est une histoire longue, passionnante et complexe. Sans rentrer dans tous les détails, nous pouvons observer plusieurs évolutions et divergences intéressantes.

SGML est d'abord conçu pour résoudre un problème lié à la représentation et la reproduction des documents. Jusqu'ici basé sur le langage Script pensé pour l'impression, les formats de document ne sont pas structuré (à part quelques exceptions comme TeX). SGML (précédé par GML), pour Standard Generalized Markup Language, introduit donc le balisage déclaratif permettant de générer des documents graphiques.

HTML est une application de SGML, plus simple et plus adapté à un affichage graphique qui ne dépend pas du système de la page imprimée. Il y a plusieurs versions successives de HTML, il faut retenir qu'il s'agit d'un ensemble d'éléments qui prennent en compte l'hypertexte. Le langage HTML est accompagné de HTTP et du système d'adresses web (les URLs).

Enfin **XML** est développé à partir des années 1990 dans un objectif de rendre plus générique un langage de balisage. XML est une suite de spécifications qui peuvent ensuite être implémentées dans des schémas, comme la TEI qui a d'ailleurs précédé XML. XML est puissant mais aussi complexe à utiliser, il est aujourd'hui très éloigné de la version actuelle de HTML. Ou disons que les choix techniques de HTML fait que ce langage est très permissif, ne respectant pas certaines prérogatives de SGML ou de XML.

Les formats légers

Le qualificatif « léger » permet de comprendre de quoi il s'agit face aux autres langages précédemment cités qui peuvent être jugés lourds. Grâce à une économie de caractères ou de signes pour baliser le texte, certains langages de balisage sont humainement lisibles (contrairement à HTML ou XML qui sont très verbeux). Avec quelques signes typographiques, des langages comme Markdown ou AsciiDoc permettent de faire de la sémantique avec un apprentissage minime.

Définition

Un format léger est un format :

- Textuel ;
- Lisible à l'œil nu, directement dans le code (sans besoin de passer par un logiciel spécialisé) ;
- Non verbeux (utilise peu de signes pour représenter la structure du texte).

La structure accompagne le texte, qui reste l'élément premier.

Exemples

- .txt — texte brut, sans balisage ;
- .md — Markdown : un format minimaliste pour structurer le texte ;
- .rst — reStructuredText, souvent utilisé dans la documentation Python ;
- .html — parfois considéré comme “semi-léger”, car il est textuel mais plus verbeux.

Une philosophie éditoriale

Trois principes :

- **Lisibilité** : un humain peut lire le fichier directement, sans le “compiler” ni l’ouvrir dans un logiciel particulier ;
- **Simplicité** : la syntaxe est minimale ; on n’a pas besoin de connaître la structure interne du document ;
- **Convertibilité** : le format léger sert souvent de point de départ vers d’autres formats (Word, HTML, PDF, LaTeX) — grâce à des outils comme Pandoc ou Quarto.

Le format texte

Un fichier au format texte est un fichier dont le code binaire (suite de 0 et de 1) se traduit par des caractères textuels uniquement (par opposition à des pixels ou des sons par exemple). Dit de manière encore plus courte : c'est un fichier qui ne contient que des caractères. (Arthur Perret, Format texte, <https://www.arthurperret.fr/cours/format-texte.html>)

Le terme *plein text*, largement utilisé, est une mauvaise traduction du terme anglais *plain text*. On préférera l'appellation *texte brut* en français.

Caractéristiques :

- Fichier texte non formaté qui ne nécessite aucun programme spécial pour être créé et ouvert ;
- Format extrêmement simple ;
- Stockage d'information en texte brut ;
- Pas de mise en forme, pas d'images, pas d'éléments multimédia ;

- Utile pour documents simples, prise de notes,...

Quelques exemples d'utilisation sur le site [The Plain Text Project](#)

Avantages :

- la **lisibilité** : contrairement aux formats binaires qui ne sont lisibles que grâce à un logiciel spécifique (comme le format .doc, et non le format .docx) ;
- la **légèreté** : les fichiers au format texte sont souvent très légers, ne dépassant pas quelques kilo-octets (car ne comporte que du texte). De là : rapide à lire par un programme, peu de place sur un support de stockage, utilisable par n'importe quel système d'exploitation ;
- la **pérennité** : un fichier informatique au format texte sera toujours lisible dans 10 ans, quand bien même les applications dédiées à son interprétation ne seront plus disponibles ;
- et dans une certaine mesure l'**interopérabilité** : un format texte est dépendant d'une spécification qu'il est possible d'implémenter ;
- enfin, le format texte est libre : pour l'ouvrir, le lire, le modifier ou le transmettre il n'est pas nécessaire de payer une licence.

.md

- Minimise l'impact des instructions de balisage dans le document, en insistant sur la lisibilité du texte par un être humain.
- Le document peut ensuite être traité par un algorithme pour un produire un export (HTML, PDF, DOCX, etc.)

Plus d'informations sur le blog du créateur de .md, John Gruber.

Markdown : exercice

Réalisez ce [tuto Markdown](#)

Stylo : mise en pratique

C'est quoi ?

- Éditeur de texte sémantique pour la rédaction scientifique ;
- Crée à l'UdeM par Marcello Vitali-Rosati et son équipe ;
- Destiné aux chercheur·e·s en sciences humaines ;
- Accessible en ligne uniquement (après création d'un compte) ;
- Pour écrire, structurer, prévisualiser, réviser, exporter (pdf, html, xml, odt, tei, etc.) ;
- Utilisation d'un langage de balisage léger (Markdown), d'un langage de description de métadonnées (YAML) et d'un langage de description bibliographique (BibTeX) ;

- Utilisation de briques logicielles existantes (et libres).

Le grand intérêt de Stylo est de permettre aux auteurs de saisir facilement leur texte dans un éditeur de texte. Ils en deviennent donc à la fois les auteurs et les encodeurs. Les concepteurs de Stylo ont souhaité construire un outil *open source* capable de garantir la bonne structuration des textes, et donc leur pérennité, leur indexation efficace, etc., sans demander aux auteurs de développer des compétences informatiques trop complexes. (Debouy, 2025, p. 79)

La documentation et des [tutos en vidéo](#).

Discussion

- Discussion autour de ([Vitali Rosati et al., 2020](#))

Exercice 1

1. Création d'un compte stylo : <https://stylo.huma-num.fr> ;
2. Reprenez le texte de Catulle vu plus haut, il se trouve aussi [ici](#) ;
3. Ajoutez-le à votre document Stylo ; ajoutez-lui un titre, éventuellement des titres de deuxième niveau, mettez des passages en italique, cherchez-lui une traduction et ajoutez des métadonnées ;
4. Partagez le document avec moi : (mathilde.verstraete@umontreal.ca) ;
5. Testez la *preview* et faites différents exports (.html, .pdf, .docx), comparez les résultats obtenus.

Alors *Stylo*, WYSIWYM ou WYSIWYG ?

Exercice 2

- Créez un nouvel article :
 - Idéalement, ajoutez un travail réalisé dans le cadre d'un autre cours
 - Si vous n'en avez pas, ajoutez n'importe quel article comprenant au moins :
 - * deux niveaux de titre,
 - * des liens,
 - * une citation,
 - * plusieurs paragraphes,
 - * une référence bibliographique
- [Envoyez-moi](#) le lien de prévisualisation lorsque vous avez terminé.

HTML

- HyperText Markup Language ;
- Inventé au tout début des années 1990 par Tim Berners-Lee, dans le même contexte que la création du World Wide Web ;
- L'idée de départ : permettre à des chercheurs de publier et relier des documents scientifiques en ligne, grâce à des liens hypertextes ;
- Premier format de publication du Web : toutes les pages web, sans exception, reposent sur du HTML (cf. `ctrl-u`) ;
- HTML sert à décrire la structure et le sens du contenu d'une page, pas sa mise en forme.

Exercice HTML

Nous allons essayer de comprendre la structure minimale d'une page html, rendez-vous ici <https://codepen.io/pen>

Template de départ (à copier-coller) :

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="utf-8" />
  <title>Représenter le texte : HTML</title>
</head>
<body>
  <!-- Travaillez ici -->
</body>
</html>
```

Ex. 1 : Structure minimale & typographie

Dans `<body>`, créez :

- un en-tête avec un titre principal `<h1>` et un sous-titre `<h2>`,
- deux paragraphes `<p>` (un court, un long),
- un mot en emphase (``) et un mot en fort (``).

Ex. 2 : Liens, images, listes

- Ajoutez une liste à puces (trois éléments) avec `...`,
- Ajoutez un lien vers un site externe qui s'ouvre dans un nouvel onglet,
- Ajoutez une image (utilise une URL publique).

Ex. 3 : Balises sémantiques & mini défi Markdown→HTML

- Plus compliqué : à partir de ce fragment Markdown, écrivez-en l'équivalent HTML

Catullus
Ad Lesbiam (V)

Vivamus mea Lesbia, atque amemus,
rumoresque senum seueriorum
omnes unius aestimemus assis!
soles occidere et redire possunt:
(5) nobis cum semel occidit breuis lux,
nox est perpetua una dormienda.

[Séance 6 : 14/10] REPRÉSENTER le texte (II): formats riches et normés

Retour sur les lectures

! Lectures obligatoires

- Coombs et al. ([1987](#))

De quoi parle-t-on ?

Certains langages de balisage permettent d'aller plus loin dans la structuration du texte :

- LaTeX :
 - langage de composition et de structuration du texte ;
 - très utilisé dans les sciences dites *dures* ;
 - langage interprété : un compilateur traduit le source (`.tex`) en une sortie (généralement `.pdf`) ;
- XML(-TEI) :
 - XML (*Extensible Markup Language*) : langage de balisage hiérarchique conçu pour représenter la structure logique d'un document ;
 - TEI (*Text Encoding Initiative*) : ensemble de balises et de recommandations pour la représentation numérique des textes (littéraires, historiques, scientifiques, etc.).
 - Utilisé dans les humanités numériques pour rendre les textes lisibles par la machine tout en conservant leur richesse sémantique et critique.

(La)TeX : un peu de contexte

- Créé par Donald E. Knuth (1938 –), mathématicien et informaticien à l'Université de Stanford et considéré comme l'un des pionniers de l'algorithlique ;

- 1977 : *The Art of Computer Programming* : mise en forme de l'éditeur ne lui plaisait pas, décide alors de développer un programme visant à rendre la composition typographique possible sur ordinateur et à modéliser le savoir-faire du typographe (6 mois → 10 ans) ;
- <τέχνη, « l'art », nom choisi car mise en page des textes techniques tout en portant sur la technique/l'art de la composition typographique.

Un an après avoir commencé, Knuth présenta l'état de son travail lors d'une conférence donnée devant les membres de l'*American Mathematical Society*, à savoir non pas un mais deux programmes : TeX pour la saisie de textes et de commandes informatiques, et METAFONT pour les fontes, un programme utilisant les courbes de Bézier pour dessiner les glyphes. Lors de cette conférence, Knuth posa également les principes suivants :

- (a) TeX était fait pour être utilisé par les auteurs des textes eux-mêmes, les mieux à même de maîtriser les normes scientifiques de leur discipline ;
- (b) Le programme, conçu dans le milieu universitaire, devait être libre et gratuit ;
- (c) Il devait fonctionner de façon strictement identique sur tous les systèmes informatiques.

(Alessi (2022), p. iv)

TeX ou LaTeX ?

- TeX : environ 900 commandes, dont 300 sont dites *primitives* (= non décomposables) = Plain TeX ;
- 1985, Leslie Lamport écrit une série de macros pour TeX et crée LaTeX : nouveaux jeux d'instructions sous la forme de commandes simples à exécuter ;
- Aujourd'hui, la très grande majorité des utilisateurs utilisent LaTeX plutôt que TeX

The LaTeX program is a special version of TeX that understands LaTeX commands. Think of LaTeX as a house built with the lumber and nails provided by TeX. You don't need lumber and nails to live in a house, but they are handy for adding an extra room.

(Lamport, 1994)

(La)TeX, concrètement

- Système de composition et de mise en page très puissant ;
- on peut TOUT faire avec LaTeX (du texte dans toutes les langues – vivantes ou anciennes –, des documents techniques ou scientifiques de qualité typographique professionnelle ; des éditions critiques ; des partitions musicales, etc.).

TeX n'est ni un traitement de texte (Knuth le nomme « typesetting system » que l'on pourrait traduire par système de composition) ni un langage de programmation compilé, c'est un langage de balisage interprété.

Esprit LaTeX

Il faut voir LaTeX comme un typographe professionnel à qui l'on confierait un manuscrit. L'homme de l'art connaît les règles de présentation et de typographie en usage. Quant à l'auteur, il ne se soucie que de son texte, laissant au typographe le soin de le composer.

(Alessi (2022), p. vi)

Le fait de ne pas s'occuper de la forme ne signifie pas que la forme n'est pas importante: au contraire! Cela signifie justement que la forme est tellement importante qu'il faut qu'un professionnel s'en occupe. LaTeX est le professionnel en question!

WYSIWYM ?

- Plus ou moins... Clairement pas WYSIWYG mais pas tout à fait WYSIWYM non plus. Mise en page *logique* plutôt que *visuelle*, selon Lamport ;
- Lozano (2013), p. xii, le qualifie de Wywsiewyg (“what you will see is exactly what you get”)

Pourquoi LaTeX ?

La réponse de Robert Alessi :

La question mérite en effet d'être posée, même si, s'agissant de travaux universitaires, il peut sembler paradoxal de préférer à un système de composition directement issu de la recherche universitaire un logiciel de secrétariat tel que le traitement de texte.

(Alessi (2022), p. vii)

- Logiciel libre : gratuit et ouvert ;

Le fait de pouvoir obtenir les sources d'un logiciel peut sembler secondaire à la plupart des utilisateurs ; il faut comprendre que c'est parce que rien n'est caché que l'amélioration de l'existant et la création d'extensions sont possibles.

(Lozano (2013), p. x)

LaTeX : Les indispensables

Tout document LaTeX se compose généralement de trois grandes parties :

- La classe du document ;
- Le préambule :
 - utilisation de packages particuliers ;

- initialisations et déclarations diverses ;
- Le corps du document (entre `\begin{document}` et `\end{document}`).

Classe du document

- Détermine la mise en page de certaines parties du document ;
- Certaines commandes ne sont disponibles que pour certaines classes : par ex., `\chapter` existe pour la classe `book` mais pas pour la classe `article`.
- Quelques classes (`\documentclass{<la-classe>}`) :
 - `article` pour un article ;
 - `report` pour un rapport de plusieurs dizaines de pages ;
 - `book` pour un livre ou une thèse ;
 - `letter` pour une lettre ;
 - etc. (dont, sa propre classe)
- `\documentclass` accepte des options, par ex.: `\documentclass[12pt,a4paper]{book}`

Préambule

- Ce qui se situe entre `\documentclass` et `\begin{document}` ;
- On y charge les *packages* (= extensions) nécessaires au document (avec `\usepackage{<nom-du-package>}` ;
- On y définit aussi les paramètres globaux du document (marges, langues, polices, style de titre, etc.) ;
- Le préambule est lu avant le contenu du document : tout ce qu'on y déclare s'applique à l'ensemble du fichier.

Corps du document

- Toujours placé entre `\begin{document}` et `\end{document}` ;
- C'est la partie principale du document, elle contient :
 - le texte,
 - les commandes de mise en forme,
 - les environnements,
 - et les références.

Quelques principes

- Ligne vierge = Saut de paragraphe (les lignes supplémentaires ne changent rien) ;
- LaTeX ignore les sauts de ligne (traités comme de simples espaces) ;
- LaTeX ignore les espaces multiples ;
- `\` = caractère “d'échappement” : ce qui suit est une séquence de contrôle ;
- `{}` = délimiteurs de groupe ;
- Caractères spéciaux : `\ $ & % # ^_ { } ~` ;
- `%` Commente la ligne qui suit ;
- `$` Symbole de début et de fin de formule (passe en mode mathématique).

Comparaison Markdown, HTML, LaTeX

Élément	Markdown	HTML	LaTeX
Liens	<code>[Lien](https://lien.com)</code>	<code>Lien</code>	<code>\href{Lien}{https://lien.com}</code>
Italique	<code>_italique_ ou *italique*</code>	<code>italique</code>	<code>\emph{italique} ou \textit{italique}</code>
Gras	<code>__gras__ ou **gras**</code>	<code>gras</code>	<code>\textbf{italique}</code>
Verbatim	<code>fonction()</code>	<code><code>fonction()</code></code>	<code>\verb fonction </code>
Images	<code>![Image](/lien/vers/img.png)</code>	<code></code>	<code>\includegraphics{img}</code>
Section	<code># Titre 1;## Titre 2</code>	<code><h1>Titre 1</h1>;<h2>Titre 2</h2></code>	<code>\section{Titre 1};\subsection{Titre 2}</code>
Citation	<code>> Citation...</code>	<code><blockquote><p>Citation...</p></blockquote></code>	<code>\begin{quote}Citation...\end{quote}</code>
Liste	<code>* item1, * item2 ou 1. item1, 2. item 2</code>	<code>item1item2 ou</code>	<code>\begin{itemize}\item item1 \item item2 \end{itemize} ou \begin{enumerate}\item item1 \item item2 \end{enumerate}</code>

(Cf. Debouy (2025), p. 91)

Un exemple de document LaTeX minimal

```
\documentclass[12pt]{article}

% --- Préambule ---
\usepackage[utf8]{inputenc}
\usepackage[french]{babel}

\title{Mon premier document en \LaTeX{}}
\author{Mathilde Verstraete}

% --- Document ---
\begin{document}
```

```
\maketitle
```

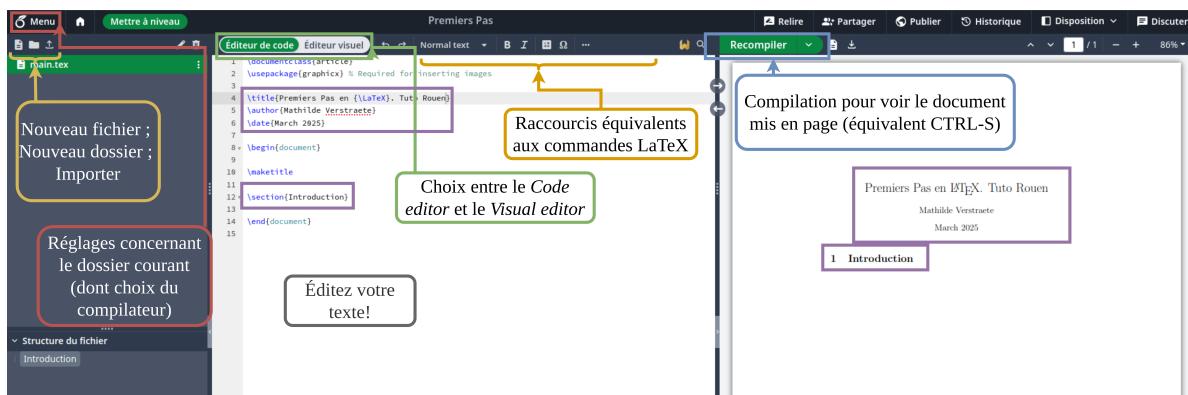
Bonjour, ceci est mon premier document \LaTeX{}!

```
\end{document}
```

Exercice : Prise en main d'OverLeaf

Overleaf est une bonne solution pour commencer à utiliser LaTeX sans avoir à installer quoi que ce soit sur votre ordinateur. Pour des travaux plus complexes, il est néanmoins conseillé d'avoir une installation locale complète (type TEXLive).

- Créez-vous un compte OverLeaf : <https://www.overleaf.com/register> ;
- Créez un nouveau projet → Projet vide → Donnez-lui un nom (“Premiers pas”);
- Familiarisez-vous avec cet environnement de travail :



- Pour compiler : CTRL-S ou “Recompiler”
- Supprimez les lignes présentes entre \documentclass{article} et \begin{document} ainsi que \maketitle et \section{Introduction} ;

Vous remarquez deux environnements distincts :

- le préambule ;
- le corps du document ;
- Dans le corps du document, écrivez votre première phrase, soit : “Bonjour tout le monde”, puis compilez ;
- Mettez votre “Bonjour” en italique, comme ceci : \emph{Bonjour} tout le monde, où :
 - \ : caractère d'échappement ;
 - emph : commande ;

- {} : groupe sur lequel agissent la ou les commandes.
- Ajoutez des commandes supplémentaires dans le préambule :
 - (a) \author{votre nom};
 - (b) \title{le titre du document};
 - (c) Compilez ! Normalement, rien ne se passe.
- Ajoutez, dans le corps du texte, un \maketitle (sans argument, il s'agit ici d'un ordre direct), compilez, et admirez !
- Testez quelques commandes supplémentaires, recompilez régulièrement afin de vous assurer que tout fonctionne :
 - (a) Ajoutez deux sections : \section{titre de la section};
 - (b) Ajoutez des sous-sections : \subsection{titre de la subsection}.
- Dans une des sous-sections, ajoutez une citation quelconque ;
- Après compilation, insérez-là dans un environnement \begin{quote}... \end{quote} :
 - (c) Ajoutez une table des matières : \tableofcontents;
 - (d) Ajoutez une note de bas de page : \footnote{texte de la note}.

Par défaut, le document est en anglais ; or, nous écrivons, depuis tout à l'heure, en français ! Les règles de typographie varient d'une langue à l'autre. Changeons cela en ajoutant, dans le préambule : \usepackage[english, french]{babel}

- Remarquez le titre Contents qui devient Table des matières ou l'ajout d'espaces fines avant le point d'exclamation et l'indication de la note de bas de page.

En résumé, on a trois types de commandes :

- (a) Les commandes sans argument (comme \maketitle);
- (b) Les commandes avec un argument (comme \footnote{});
- (c) Les commandes avec un argument et une ou des options (comme \usepackage[lang1, lang2, langn]{babel}).

LaTeX : Pour aller plus loin

- <https://www.learnlatex.org/> propose d'excellents tutoriels.

Et après Overleaf ?

Pour des travaux plus importants, il est conseillé de disposer d'une installation complète de LaTeX sur sa machine. Pour ce faire, un espace d'environ **6 Go** sur son disque est nécessaire (c'est lourd, et l'installation prend du temps).

Windows

- Cet utilitaire <http://mirror.ctan.org/systems/texlive/tlnet/install-tl-windows.exe> lancera l'installation. Sauvegardez-le temporairement sur votre bureau. Une fois l'installation terminée, il pourra être déplacé dans la corbeille. Ensuite, deux installations sont possibles :
 - en mode utilisateur : double-clic sur le fichier `install-tl-windows.exe` ;
 - en mode administrateur (pour que tous les utilisateurs de la machine puissent utiliser TeXLive) : clic droit sur le fichier `install-tl-windows.exe` > exécuter en tant qu'administrateur.
- Dans les deux cas : cliquer sur **Install** (lance l'installation : prévoir un bon réseau et 1 à 3h de temps).
- Redémarrez l'ordinateur

Il vous faudra un éditeur de texte : VSCodium fonctionnera, mais [TeXmaker](#) sera plus adapté.

Mac OS

- La distribution s'appelle MacTeX et est disponible à l'adresse <http://www.tug.org/mactex/mactex-download.html> ;
- Cliquez sur le lien `MacTeX.pkg` (prévoir un bon réseau et 1 à 3h de temps) ;
- Dans le répertoire Téléchargements, double clic sur `MacTeX.pkg` ;
- Sur toutes les fenêtres qui s'ouvrent, cliquez sur **Continuer** puis **Accepter**. Il vous faudra finalement saisir votre mot de passe puis **Installer le logiciel**.

Il vous faudra un éditeur de texte : VSCodium fonctionnera, mais [TeXmaker](#) sera plus adapté.

XML-TEI

XML

- *eXtensible Markup Language* ;
- Développé à la fin des années 1990 par le W3C, dans le but de simplifier le SGML (*Standard Generalized Markup Language*) et de rendre le balisage textuel plus universel et interopérable ;
- Consiste en une suite de spécifications implémentables dans des *schémas* (comme la TEI – qui a d'ailleurs précédé XML) :

- XML est un métiformat : il ne définit pas un ensemble de balises fixe, mais une syntaxe générique permettant de créer de nouveaux langages structurés (schémas) ;
- Repose sur le principe de hiérarchie et de structure logique, lisible par l'humain comme par la machine ;
- Puissant mais complexe à utiliser.

Un exemple très minimalisté:

```
<mondocument>
  <titre>Un exemple d'XML</titre>
  <texte>
    <paragraphe>
      Ce texte est un exemple d'XML.
    </paragraphe>
  </texte>
</mondocument>
```

TEI

- *Text Encoding Initiative* :
- Les spécifications : <https://www.tei-c.org/> ;
- Des exemples : <https://teibyexample.org/> ;
- Consortium international fondé en 1987, rassemblant chercheurs, éditeurs et institutions des humanités numériques ;
- Objectif : définir des recommandations normalisées pour l'encodage de textes, afin d'assurer leur interopérabilité et leur pérennité ;
- Repose aujourd'hui sur XML, après avoir d'abord utilisé SGML ;
- La TEI définit un ensemble de balises et de modules permettant de décrire la structure logique, le contenu et les annotations d'un texte (titres, notes, apparat critique, correspondance, etc.) ;
- On ne parle pas de format, mais d'une famille de schémas dérivés d'un modèle commun.

Un exemple: une lettre de Juliette Drouet:

```
<teiCorpus xml:id="jd.entry.7912">
  <TEI xml:id="jd.entry.7912.01">
    <teiHeader>
      <fileDesc>
        <titleStmt>
          <title>Title</title>
          <respStmt>
            <resp>Rétroconversion Spip vers XML-TEI</resp>
```

```
<persName>Josselin Morvan</persName>
</respStmt>
<respStmt>
    <resp>Validation des fichiers XML-TEI</resp>
    <persName>Élisa Barthélémy</persName>
    <persName>Hélène Hôte</persName>
</respStmt>
<respStmt>
    <resp>Transcription</resp>
    <persName>Apolline Ponthieux</persName>
</respStmt>
<respStmt>
    <resp>Contrôle</resp>
    <persName>Florence Naugrette</persName>
</respStmt>
</titleStmt>
<sourceDesc>
    <msDesc>
        <msIdentifier>
            <repository>BnF, MSS, NAF 16400, f. 246</repository>
        </msIdentifier>
    </msDesc>
</sourceDesc>
</fileDesc>
<profileDesc>
    <correspDesc>
        <correspAction type="written">
            <persName>Juliette Drouet</persName>
            <date when="1879-10-14">14 octobre 1879</date>
            <placeName>Paris</placeName>
        </correspAction>
        <correspAction type="received">
            <persName>Victor Hugo</persName>
        </correspAction>
    </correspDesc>
</profileDesc>
</teiHeader>
<text>
    <body>
        <div type="letter">
            <opener>
                <dateline rend="right">
```

Paris, 14 octobre
<supplied>18</supplied>79, 1 h. du matin
</dateline>
</opener>
<p>
Cher bien-aimé, je te porte le premier toast avant tous ceux
qu'on te portera ce soir
<note type="footnote">
Victor Hugo se trouve au Grand-Hôtel, pour le banquet
qu'il a organisé à l'occasion de la centième
représentation de l'adaptation de <emph>Notre-Dame de Paris</emph>
par Paul Meurice, au Théâtre des Nations.
Cet événement rassemble des personnalités éminentes
des milieux littéraire, artistique et journalistique.
</note>
qui ne seront d'ailleurs, quelque
<note type="manuscriptologique">
« quelques ».
</note>
nombreux et quelque
<note type="manuscriptologique">
« quelques ».
</note>
enthousiastes qu'ils soient, que l'écho affaibli du mien.
Pense à moi qui te souris et qui te bénis
et aime-moi, moi, qui t'adore.
<lb/>
Je ne sais pas si je me coucherai avant ton retour car
il me semble que mon cœur te garde mieux les yeux ouverts.
Je crains pour toi les excitations inévitables de la gloire
versée à plein bord par tous ceux qui sont là au risque
de te donner trop d'émotion à la fois.
Cher adoré, tu ne sais pas combien je t'aime !
Tu ne le sauras que lorsque tu verras mon âme tout entière,
alors tu comprendras tout ce que ma sollicitude pour toi
a de vénérable et de sublime devant Dieu.
<lb/>
Je baise ton divin front.
</p>
<div type="address">
<p>
<supplied>Adresse</supplied>

```

<lb/>
Monsieur Victor Hugo
</p>
</div>
</div>
</body>
</text>
</TEI>

```

Exercice : Écrire en TEI-XML (avec Leaf-Writer)

LEAF est un outil éditorial pour la collaboration et la publication ouvertes. Il fournit des outils basés sur le web et des espaces en ligne permettant à un plus grand nombre de chercheurs, d'enseignants et d'étudiants de participer à la production collaborative de connaissances numériques : créer, publier, préserver, relier et partager des données sur le web.

LEAF-Writer est un éditeur sémantique basé sur le web qui permet de baliser des documents en utilisant une combinaison de XML et de RDF. Il prend en charge la TEI pour l'encodage du contenu scientifique, y compris des schémas personnalisés, et crée en option des annotations mettant à profit le Web des données ouvertes et liées.

- Voir [la documentation](#) et un [tutoriel](#).

Extra : Quelques outils des Humanités numériques⁷

Hypothes.is

Annotate the web, with anyone, anywhere.

Hypothes.is est un outil d'annotation pour le Web, il est basé sur les standards du Web et permet de créer des conversations.

- fonction(s) : annoter des pages web ;
- usage(s) : créer des conversations sur des documents ;
- création : 2011, Dan Whaley ;
- mise à jour : développements continus ;
- changement(s) induit(s) : implémentation d'un standard pour annoter tout document numérique.

⁷Cette section est tirée du cours HNU2000-A22 donné par A. Fauchié.

Pour bien appréhender Hypothes.is il faut comprendre le fonctionnement du Web en détail ainsi que les enjeux des standards.

The image contains three screenshots of the Hypothes.is interface, illustrating how users can annotate and discuss online content. The first screenshot shows a general annotation interface with a sidebar for navigation. The second and third screenshots show specific annotations on a seminar slide about 'Autonomie et présence artificielle' and a Beam video, respectively. These annotations include text highlights and responses from other users.

Stylo

Stylo est un éditeur de texte sémantique pour les sciences humaines.

Stylo est un outil conçu pour repenser le processus d'écriture et d'édition des revues savantes en sciences humaines et sociales. En tant qu'éditeur de texte sémantique WYSIWYM (*What You See Is What You Mean*) pour les sciences humaines, il vise à améliorer la chaîne de publication académique.

- fonction(s) : écrire et structurer du texte dans une interface web ;
- usage(s) : créer des articles scientifiques en respectant des protocoles d'édition ;
- création : 2008, Chaire de recherche du Canada sur les écritures numériques ;
- mise à jour : plusieurs fois par an ;
- changement(s) induit(s) : repenser l'écriture scientifique et les chaînes de publication académique.

Stylo est basé sur des standards et des protocoles, c'est aussi un outil qui fonctionne grâce au balisage sémantique et grâce à d'autres outils libres.

Zotero

Your personal research assistant

Zotero est un logiciel permettant de récolter et d'organiser des références bibliographiques, mais aussi un service en ligne permettant de sauvegarder et de partager des références bibliographiques.

- fonction(s) : enregistrer, organiser, sauvegarder et partager des références bibliographiques ;
- usage(s) : création et gestion de références bibliographiques à usage universitaire et éditorial ;
- création : 2006, Roy Rosenzweig Center for History and New Media ;
- mise à jour : évolutions très régulières ;

- changement(s) induit(s) : outil de gestion de références bibliographiques libre et gratuit.

Zotero est un outil qui semble parfois trop avancé pour une question si simple que la création de références bibliographiques ou de bibliographies. Dans les faits Zotero se trouve être un outil incontournable, mais pour arriver à cette conclusion il faut appréhender les questions de contraintes éditoriales en milieu académique, ainsi que la définition d'une bibliographie structurée (ou de toute donnée structurée en général).

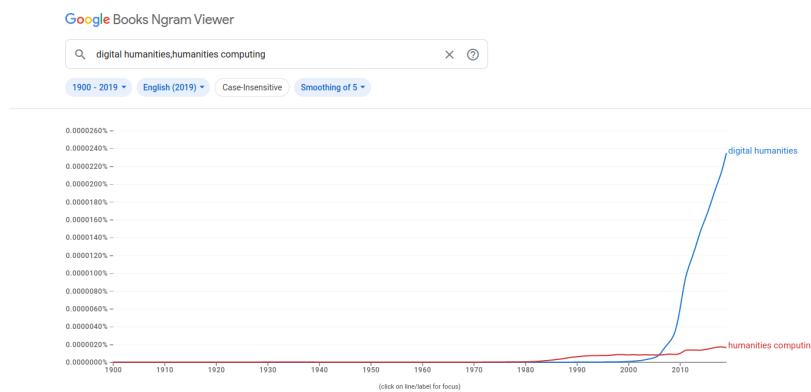
Ngram Viewer

Ngram Viewer est une application linguistique proposée par Google, permettant d'observer l'évolution de la fréquence d'un ou de plusieurs mots ou groupes de mots à travers le temps dans les sources imprimées. ([Wikipédia](#))

Ngram Viewer est un pur produit de Google : il s'agit d'interroger les livres numérisés par Google autant que d'utiliser des algorithmes de recherche produits par Google.

- fonction(s) : recherche, comparaison et affichage de termes dans un corpus défini ;
- usage(s) : comparaison de termes dans l'un des plus importants corpus de livres numérisés ;
- création : 2010, Google ;
- mise à jour : février 2020 (pas possible d'interroger les livres publiés après 2019) ;
- changement(s) induit(s) : comparer facilement un immense corpus, avec une visualisation graphique.

Il faut découvrir comment un livre est numérisé pour prendre la mesure d'une telle application, ainsi que comprendre comment un moteur de recherche permet de parser une quantité de données aussi importante.



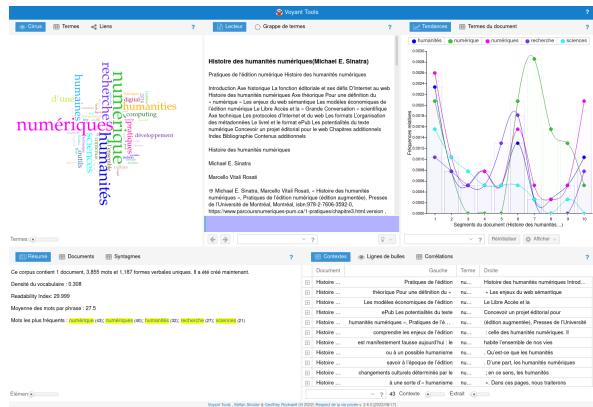
Voyant Tools

Voyant Tools est un environnement en ligne de lecture et d'analyse de textes numériques.

Voyant Tools est un projet universitaire proposant un environnement de lecture et d'analyse de texte sur le Web. C'est un outil pédagogique permettant de comprendre ce qu'est la fouille de texte.

- fonction(s) : extraction, calcul de la fréquence, production de statistiques, etc. ;
 - usage(s) : outil d'analyse de texte ;
 - création : 2003, Stéfan Sinclair et Geoffrey Rockwell ;
 - mise à jour : développements réguliers ;
 - changement(s) induit(s) : disposer d'outils d'analyse de texte accessibles.

La question du logiciel libre est ici importante : comment un programme informatique ou une application web peut être créée et maintenue ? Cette application permet de comprendre les enjeux (complexes) de la fouille de texte.



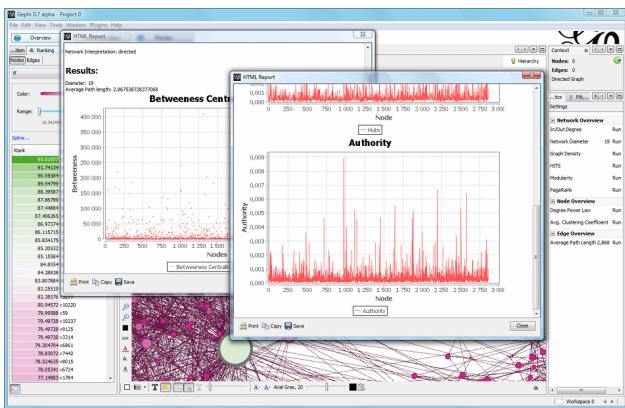
Gephi

Gephi is the leading visualization and exploration software for all kinds of graphs and networks. Gephi is open-source and free.

Gephi est un logiciel de visualisation et d'exploration de réseaux, très utilisé en histoire ou dans le domaine du journalisme.

- fonction(s) : visualiser des graphes en appliquant un certain nombre de calculs ;
 - usage(s) : donner une représentation graphique de réseaux ;
 - création : 2008, notamment par Sébastien Heymann ;
 - mise à jour : une ou plusieurs fois par an ;
 - changement(s) induit(s) : visualiser des informations impossibles à.

Gephi permet d'appréhender la théorie des graphes et de comprendre pourquoi la visualisation a une place prépondérante dans certaines disciplines.



Qu'est-ce qu'une étude de cas ?

- analyse d'un objet/projet ;
- comprendre les implications scientifiques, techniques et fonctionnelles ;
- description et regard critique ;
- expérimenter.

L'étude de cas est une analyse d'un objet ou d'un projet permettant d'en comprendre les implications scientifiques, techniques et fonctionnelles. À la fois description référencée et sourcée, et regard critique, une étude de cas est un travail synthétique qui répond à des questions comme « qui ? », « comment ? », « pourquoi ? » ou « quand ? ». En décrivant les origines et le fonctionnement d'un objet, l'analyse de cas vise à expliquer l'intention initiale ainsi que de mesurer la pertinence de la démarche dans un contexte.

C'est un exercice qui requiert de la méthode, ainsi que des recherches et des manipulations. Dans le cadre de ce cours en partie pratique, il est important de tester soi-même l'objet étudié afin d'en tirer une expérience.

[Séance 9 : 11/11] ÉCRIRE & PUBLIER

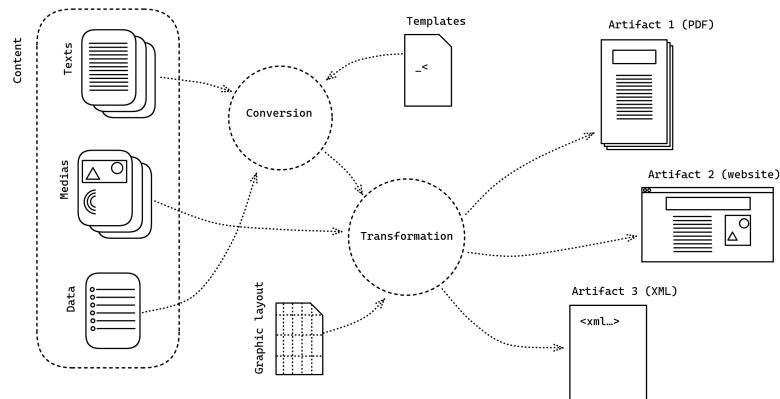
Retour sur les lectures

! Lectures obligatoires

- Fauchié & Audin (2023)

Le Single Source Publishing

L'expression *single source publishing*, ou publication à source unique en français, désigne le fait de produire plusieurs formats ou artefacts différents à partir d'une seule et même source, dans le cadre d'une démarche de publication.



Une seule source

- Une source unique ;
- Pour éviter les confusions ;
- Pour gagner en énergie ;
- Différents types d'interventions.

Modéliser les sorties

- Un format de sortie = un modèle/template spécifique ;
- Répartir chaque information ;
- Prendre en compte un certain nombre de contraintes.

Quelques exemples

- [Lodel](#) ;
- [Métopes](#) ;
- [Scalar](#) ;
- [Manifold](#) ;
- [Quire](#).

Une approche modulaire

- *Chaîne d'édition classique* : Word + InDesign ;
- *Chaîne d'édition modulaire* : un ou plusieurs modules remplissent une ou plusieurs fonctions.

Les chaînes d'édition classiques reposent sur des logiciels par défaut, c'est-à-dire que ceux-ci sont utilisés faute d'application mieux adaptées.

Quelques exemples

- *Style* : Markdown, YAML et BibTeX pour les données ; Pandoc et LaTeX pour la génération des formats ;
- *Quire* (article) : Markdown, YAML et JSON pour les données ; Hugo et PrinceXML pour la génération des formats ;
- [Abrupt](#) : Markdown, HTML, YAML ; Make, Pandoc et LaTeX pour la génération des formats.

Un exemple d'outil : [Quarto](#)

- Outil open-source de création et de publication de documents scientifiques (rapports, présentations, livres, sites, etc.) ;
- Permet de combiner du .md avec du code dans plusieurs langages (R, Python, Julia, etc.) ;
- Production de documents dans différents formats : HTML, PDF, Word, diaporamas interactifs,...

Researchers shouldn't need to choose between LaTeX, MS Word, and HTML but rather be able to author documents that target all of them at the same time. ([Quarto](#))

Make reproducible research and publications the norm rather than the exception. ([Quarto](#))

Fonctionnement : .md (+ syntaxe “quarto” incluant code et options de formatage) + .yaml (dans le .md ou à part) + pandoc = vos formats de sortie

Note sur Pandoc

- Logiciel libre en ligne de commande pour la conversion de fichiers texte ;
- John MacFarlane, licence GPL ;
- « Couteau suisse de l'édition » ([Perret 2018](#)) ;
- Documentation : <https://pandoc.org/MANUAL.html>.

Exemple (simple) :

```
pandoc --citeproc -f markdown -t epub TexteAvant.md -o TexteApres.epub
```

Traduction :

cher pandoc avec l'option de citation, fais la conversion, depuis (from) le Markdown vers (to) le ePuB, de mon fichier de départ (TexteAvant .md) vers mon fichier de destination (out) (TexteApres .epub)

Exemple (moins simple) :

```
pandoc --citeproc --template=Template.latex -f markdown -t latex TexteAvant.md  
-o TexteApres.tex
```

Traduction :

cher pandoc avec l'option citation, en utilisant et mon template (qui est le fichier Template .latex), fais la conversion, depuis (from) le Markdown vers (to) le Latex, de mon fichier de départ (TexteAvant .md) vers mon fichier de destination (out) (TexteApres .tex)

Manipuler quarto

1. Installez Quarto : <https://quarto.org/docs/get-started/> (et choisissez VSCode comme outil) ;
2. Faites le tuto *Hello, Quarto* : <https://quarto.org/docs/get-started/hello/text-editor.html> :
 - Vous pouvez passer les parties où il est question de code informatique, nous nous concentrerons sur le texte ;
3. Faites le tuto *Authoring* : <https://quarto.org/docs/get-started/hello/text-editor.html> ;
4. Créez une présentation : rendez-vous dans un nouveau répertoire consacré au cours et suivez les informations décrites [dans la documentation](#) :
 1. Creating Slides ;
 2. Incremental Lists ;
 3. Multiple Columns ;
 4. Content Overflow ;
 5. Speaker Notes ;

6. Themes ;
7. Asides & Footnotes.

La gestion de versions

Cela vous rappelle-t-il quelque chose ? `mon-fichier-v2-relu-2021-12-09-final-ok-okok.txt`

Il est utile d'utiliser un système de contrôle de version (ou VCS) en anglais afin de suivre toutes les modifications du code saisi pour produire un texte.

Le versionnement entend répondre à plusieurs objectifs :

- Enregistrer un ou plusieurs fichiers ;
- Garder une trace des versions des fichiers ;
- Naviguer dans l'historique des versions ;
- (Le tout à plusieurs).

Décentraliser pour mieux travailler

Plusieurs principes inhérents à une bonne gestion collective de fichiers informatiques :

- Ne pas centraliser la sauvegarde ;
- Cloner facilement = copier le dossier de travail avec l'ensemble de l'historique ;
- Travailler hors connexion ;
- Créer des versions parallèles d'un projet.

Les différents systèmes de contrôle de version

- Systèmes locaux :
 - Modifications enregistrées dans une base de données ;
 - Seules sont enregistrées les différences dans un seul fichier annexe, soit : fichier initial + fichier de différences ;
- Systèmes centralisés :
 - Idem, mais les bases de données se trouvent sur un serveur ;
 - Possibilités de travail en groupe ;
 - Si panne de serveur, base de données inaccessible, impossibilité de travailler ;
- Systèmes distribués :
 - Un ensemble de composants indépendants situés sur des machines différentes qui partagent des messages entre eux afin d'atteindre des objectifs communs ;

- Le travail se fait sur une version autonome et dupliquée de tous les fichiers présents sur le serveur ;
- Chacun travaille sur ses propres fichiers (même hors ligne) ;
- Une fois une portion de travail terminée, envoi sur le serveur d'un nouvel état de la base de données que les autres pourront alors synchroniser ;
- Possibilité d'accéder à un dépôt à partir de plusieurs emplacements ;
- Git.

Introduction à Git

- Un logiciel de gestion de versions parmi d'autres (pour les fichiers en plein texte) ;
- Système distribué ;
- Avantage : plutôt que de considérer les *différences* entre modifications des fichiers, il sauvegarde des *instantanés* complets.

Versionnement

chaque fois qu'on enregistre (*commit*), ou qu'on sauvegarde l'état de notre projet dans Git, Git prend un instantané de tous les fichiers, en sauvegardant une référence à l'instantané. Git traite donc nos fichiers comme une série d'instantanés dans le temps. Pour être plus efficace, si Git reconnaît que certains fichiers n'ont pas changé depuis l'instantané précédent, Git ne les récupère pas à nouveau, mais crée simplement une collation vers le fichier précédent qui a déjà été sauvegardé.

Exemple ([Debouy, 2025, p. 82-83](#))

Soit un fichier `recherche.txt` :

```
PREMIÈRE PARTIE
=====
COMBRAY
-----
I
Longtemps, je me suis couché très tôt.
Parfois, à peine ma bougie éteinte,
mes yeux se fermaient si vite que j'avais
juste le temps de me dire : « Je m'endors. »
```

Que je modifie de la sorte :

PREMIÈRE PARTIE

COMBRAY

I

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. »

En analysant les différences entre les deux états de `recherche.txt`, j'obtiendrai :

```
--- recherche.txt    2018-10-03 12:35:05.848903296 +0200
+++ recherche.txt    2018-10-03 12:31:04.292356175 +0200
@@ -6,6 +6,6 @@

```

I

-Longtemps, je me suis couché très tôt. Parfois, à peine ma
-bougie éteinte, mes yeux se fermaient si vite que j'avais juste le
+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.»

Que j'interprète de la sorte :

- l. 3 : aucune modification avant la l. 6 du fichier original (`recherche.txt`) ;
- ll. 5 et 10 : lignes non modifiées (= contexte, ll. 7 et 11 du fichier original) ;
- ll. 6 et 7 : précédées du signe -, remplacées par les ...
- ... ll. 8 et 9, précédées quant à elles du signe +.

Quelques principes

- Suivre un projet ;
- Développements/évolutions non linéaires (branches) :
 - travail simultané sur plusieurs *branches* : une branche principale et des branches secondaires ;
 - Ex. :

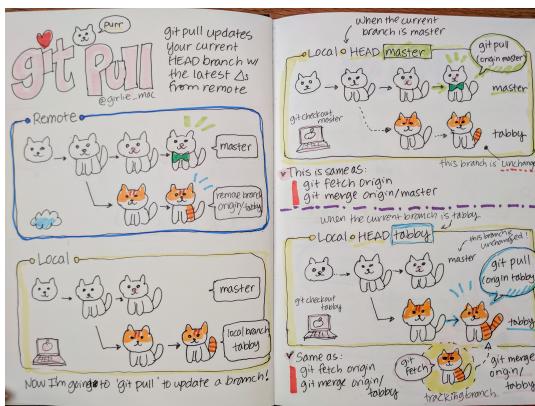
- à un moment donné, on souhaite revenir sur une page complexe que l'on a rédigée et y apporter des modifications et on voudrait ne rien perdre de la version originale. On ouvre alors une *branche secondaire* dans laquelle on modifie à souhait la page. Durant tout le travail, sur la branche secondaire, le travail enregistré dans la branche principale n'est pas altéré. Une fois terminé le travail sur la branche secondaire, on peut l'abandonner ou bien, si le résultat est satisfaisant, le conserver. Dans ce cas, on demande au système de contrôle de version de fusionner dans la branche principale la branche secondaire sur laquelle on a travaillé. Puis on continue le travail dans la branche principale, après avoir effacé la branche secondaire, tout en sachant qu'à tout moment, toutes les étapes de ces modifications peuvent être retrouvées. (Debouy, 2025, p. 81)

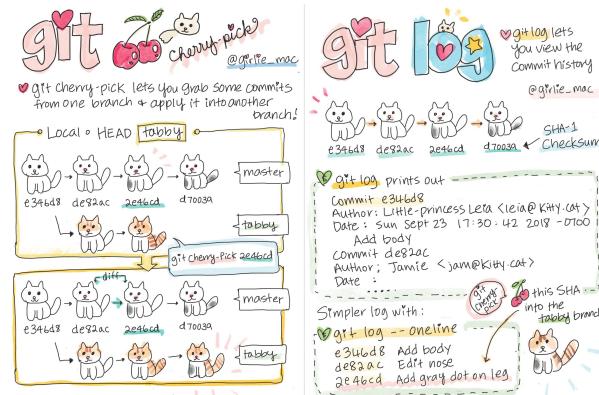
- Autant de branches qu'on le souhaite ;

- Historique général et particulier ;
- Système de gestion de versions distribué ;
- Complexité relative ;
- Logiciels/plateformes qui simplifient l'usage.

Git est pensé pour versionner un projet, seul·e ou à plusieurs. Si Git permet de versionner des fichiers, il ne faut pas oublier que l'objectif final est bien de suivre un projet dans son ensemble et pas forcément un fichier en particulier

Une illustration





(source)

Quelques termes

- **Dépôt** : ensemble des fichiers versionnés ;
- **Commit** : enregistrement des changements dans un dépôt :
 - Pour comprendre comment fonctionne Git il faut comprendre ce qu'est un commit : ce n'est pas un enregistrement classique, mais l'état du projet tout entier après une série de modifications sur un ou plusieurs fichiers ;
- **Fork** : divergence d'un projet, une copie qui va vivre sa propre vie (ou presque) :
 - En forkant on s'approprie le projet, si besoin on pourra soumettre des modifications à la version originale tout en travaillant sur notre version ;
- **Cloner** : copier un projet chez soi pour pouvoir y contribuer :
 - En clonant le projet je conserve le projet original, mais je risque de perturber les développements ;
- **Conflit** : lorsque l'on tente de réunir voir de fusionner plusieurs versions d'un même projet.

Un commit est une série d'informations :

- un identifiant ;
- une date ;
- un message ;
- une liste de modifications associées : les fichiers modifiés.

En somme

- Bien configurer Git ;
- Git ≠ GitHub/GitLab ;
- La bible : <https://git-scm.com/book/fr/v2> ;
- Il est presque impossible de supprimer quelque chose.

Manipulation : Git et GitHub.

Les commandes usuelles

- `git init` : initialiser un dossier ;
- `git status` : voir l'état du projet ;
- `git log` : afficher l'historique de la branche actuelle ;
- `git add` : ajouter un fichier dans l'index avant de commiter ;
- `git commit` : déclarer des modifications ;
- `git branch` : créer une nouvelle branche ;
- `git checkout` : pour basculer sur une branche ;
- `git push` : envoyer les modifications sur un dépôt distant ;
- `git fetch` : récupérer les modifications d'un dépôt distant ;
- `git pull` : récupérer les modifications d'un dépôt distant et les fusionner avec le dépôt local.

Pour afficher un historique plus détaillé : `git log --decorate=full --raw`, ou pour afficher l'arbre des branches : `git log --branches --remotes --tags --graph --oneline --decorate --pretty=format:"%h - %ar - %s"`

Installation de Git

! Important

Git ≠ GitHub/GitLab

- Installer Git : <https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-Installation-de-Git>

- Le plus simple, pour ce cours, est de l'installer avec les installations Github, soit,
 - * Pour MacOS : <https://desktop.github.com/> ;
 - * Pour Windows : <https://windows.github.com/> ou <https://desktop.github.com/> ;
- Initialiser Git (peut-être, selon l'installation ci-dessus) : lui donner votre nom et votre email :

```
git config --global user.name "John Doe"
git config --global user.email johndoe@example.com
```

Pour vérifier si git est bien installé : `git --version`

Clés SSH

- *Secure Socket Shell* ;
- Protocole de cryptage ;
- Pour faire interagir mon ordinateur (client) avec un serveur en sécurité ;

Prise en main de Git

- Créez un répertoire de travail dans le dossier dédié au cours, par ex :
 - `cd Documents/HNU2000` ;
 - `mkdir Seance_9` ;
 - `cd Seance_9` ;
 - `git init` : on demande à Git de suivre le contenu de ce répertoire ;
 - Vous devriez obtenir comme “réponse” :
 - * Dépôt Git vide initialisé dans `/home/votre-nom/Documents/HNU2000/Seance_9/.git/`
 - * la base de données sera maintenue dans un répertoire caché `.git` lui-même à l'intérieur du répertoire `Seance_9`.
- Ouvrez VSCode (en veillant à bien être dans le même répertoire) et créez un document `recherche.txt` et ajoutez-y le texte du premier état du fichier donné en exemple plus haut.
- Sauvez puis, dans votre terminal, faites un `git status` ; la réponse devrait être :

```
[votre-nom Seance_9]$ git status
Sur la branche master

Validation initiale

Fichiers non suivis:
  (utilisez "git add <fichier>..." pour inclure dans ce qui sera validé)
```

```
recherche.txt
```

aucune modification ajoutée à la validation mais des fichiers non suivis sont présents (utilisez "git add" pour les suivre)

Voici comment interpréter ce message :

- l. 2 : par défaut, la *branche principale* est appelée par Git **master** (ou **main**, selon les configs) ;
- l. 6 sq. : liste de fichiers trouvés dans le répertoire + indique qu'on ne lui a pas (encore) demandé de les *suivre* ;
 - ! Git ne suivra que les fichiers qu'on lui a demandé d'indexer !
- l. 7 : instructions pour indexer le fichier qui a été trouvé: `git add <fichier>`

Ajoutons le fichier et demandons à Git un nouveau rapport :

- `git add recherche.txt` : place le fichier `recherche.txt` dans la zone d'index de Git (la zone de travail, ou *staging area*) ;
- `git status` ;
- Vous devriez obtenir en réponse :

```
Sur la branche master
```

```
Validation initiale
```

Modifications qui seront validées :

(utilisez "git rm --cached <fichier>..." pour désindexer)

```
nouveau fichier : recherche.txt
```

Le fichier est désormais indexé, mais pas enregistré dans la base de données :

- `git commit <fichier> -m "<message>"` ou plutôt `git commit recherche.txt -m "version initiale"` ;
- Avec, en réponse :

```
[master (commit racine) fa1ec00] version initiale
1 file changed, 11 insertions(+)
create mode 100644 recherche.txt
```

Modifions maintenant le fichier avec la deuxième version donnée en exemple plus haut ;

- sauvez ;
- `git diff` : affiche les différences entre la version actuelle des fichiers et leur dernière version enregistrée (par un `git commit`) ;
- la réponse devrait être :

```
diff --git a/recherche.txt b/recherche.txt
index 3baf502..f230132 100644
--- a/recherche.txt
+++ b/recherche.txt
@@ -6,6 +6,6 @@ COMBRAY
```

I

-Longtemps, je me suis couché très tôt. Parfois, à peine ma
 -bougie éteinte, mes yeux se fermaient si vite que j'avais juste le
 +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.»

Pour enregistrer ces informations :

- `git commit recherche.txt -m "nouvelle version de l'incipit"`
- Réponse :

```
[master 83b6c3e] nouvelle version de l'incipit
 1 file changed, 2 insertions(+), 2 deletions(-)
```

Demandons à Git de nous fournir un extrait de son journal :

- `git log` ;
- Réponse :

```
commit 83b6c3e6dad72116eac5ce7d1ba70968e4e57ebb
Author: Votre-Nom <votre-adresse@email.com>
Date:   Wed Oct 3 15:05:32 2018 +0200

  nouvelle version de l'incipit

commit fa1ec001efdca9c69cc768dc9cf83706bdb6e603
Author: Votre-Nom <votre-adresse@email.com>
Date:   Wed Oct 3 14:49:10 2018 +0200

  version initiale
```

- Analyse :
 - ll. 1 et 7 : à chaque *commit* est associé un numéro de registre, en notation hexadécimale, formé d'une séquence de 40 caractères allant de 0 à 9 et de a à f.
 - * Cette chaîne est appelée *hash* ;
 - * un identifiant pour un commit ;
 - * ex. : 24b9da6552252987aa493b52f8696cd6d3b00373

Tuto

Réalisons ensemble quelques uns des tutos disponibles sur le site : https://learngitbranching.js.org/?locale=fr_FR&demo=

- Regarder la démo ;
- Tuto Principal :
 - Les 4 exercices de la séquence d'introduction ;
- Tuto Remote :
 - Les exercices 1 à 6 de la séquence Push & Pull

Résumé des commandes

- `git config --global` : paramétrage initial de Git ;
- `git init` : initialisation de Git dans un nouveau répertoire ;
- `git status` : demande à Git un *rappor d'état* ;
- `git add` : indexe des fichiers dans la zone de travail. Seuls les fichiers indexés seront suivis par Git ;
- `git commit <fichier> -m "<message>"` : enregistre dans la base de données de Git les versions modifiées des fichiers sous forme d'instantanés ;
- `git diff` : montre les différences entre la version actuelle des fichiers et leur dernière version enregistrée par `git commit` ;
- `git log` : affiche le journal de Git.

Exercice

Étape 1 — Créer et initialiser un dépôt local :

- créer un dossier : `mkdir bac-a-sable` ;
- aller dans ce dossier : `cd bac-a-sable` ;
- créer un fichier : `echo "Mon texte" >fichier-01.txt` ;
- initialiser un dépôt git : `git init` ;
- ajouter le fichier dans l'index : `git add fichier-01.txt` ;

- commiter : `git commit -m "Premier commit : ajout du fichier-01.txt";`
- afficher l'historique : `git log.`

Étape 2 — Synchroniser avec GitHub :

1. Créer un dépôt vide sur GitHub :
 1. Se rendre sur <https://github.com> et s'y connecter ;
 2. Créer un “New repository” ;
 3. Nommer le dépôt : `bac-a-sable` ;
 4. N’ajoutez pas de Readme ;
 5. Copier l’adresse SSH du dépôt (elle ressemble à `git@github.com:nom-utilisateur/bac-a-sable.git`).
2. Lier le dépôt local à GitHub :
 1. Dans le terminal, à la racine du dossier `bac-a-sable` : `git remote add origin git@github.com:nom-utilisateur/bac-a-sable.git;`
 2. Il sera sans doute nécessaire de configurer votre clé SSH à cette étape ;
3. Vérifier la connexion : `git remote -v` ;
4. Envoyer le projet sur GitHub : `git push -u origin main` ;
 - Si la branche s’appelle `master` il faudra la renommer :
 - `git branch -M main` ;
 - `git push -u origin main` ;

Étape 3 — Synchroniser les changements :

- Récupérer les changements faits sur GitHub : `git pull` ;
 - (`git pull = git fetch + git merge`) ;
- Récupérer les infos du dépôt distant sans fusionner : `git fetch` ; - Intégrer les nouvelles modifications : `git rebase` ; - Envoyer les commits locaux vers GitHub : `git push`.

Exercice : résoudre un conflit

- Créer une nouvelle branche appelée `modifs` : `git checkout -b modifs` ;
- Vous êtes désormais sur cette nouvelle branche ;
- Modifier le fichier, par exemple : `echo "Autre texte hop là" >fichier-01.txt` ;
- Enregistrer vos modifications dans Git : `git commit -a -m "révision de la première ligne"` ;
- Retourner sur la branche principale : `git checkout master` (ou `git checkout main` selon votre configuration) ;
- Effectuer une nouvelle modification : `echo "Ceci est mon texte" >fichier-01.txt` ;
- Enregistrer vos modifications : `git commit -a -m "réécriture"` ;

- Tenter de fusionner les deux branches : `git merge modifs` ;

Vous devez avoir un conflit !

Vous devez ouvrir le fichier pour résoudre le conflit :

- Dans un éditeur de texte comme VSCode, des options vont vous être proposées pour choisir la version souhaitée ;
 - C'est à vous de conserver manuellement ce qui vous intéresse
- Une fois ces modifications faites, vous devez enregistrer le fichier (`git add`), et commiter tout cela (`git commit`) ;
- Effectuer un `git status` pour s'assurer que tout est en ordre !
- Si vous arrivez pas à résoudre les conflits et vous voulez retourner à la situation précédente : `git merge --abort` ;

Références

- Alessi, R. (2022). *Initiation à LaTeX*.
- André, J. (1990). *Petites Leçons de Typographie*. Éditions du jobet.
- Bermès, E. (2023). Trente ans de numérique à la BnF. *Balisages. La revue de recherche de l'Enssib*, 6. <https://doi.org/10.35562/balisages.1099>
- Bot, J.-M. L. (2012). Contribution à l'histoire d'un lieu commun : l'attribution à Chateaubriand de la phrase « les forêts précèdent les peuples, les déserts les suivent ». *Socio-logos. Revue de l'association française de sociologie*, 7. <https://doi.org/10.4000/socio-logos.2634>
- Burdick, A., Drucker, J., Lunenfeld, P., Presner, T., & Schnapp, J. (Éds.). (2012). Humanities to Digital Humanities. In *Digital_Humanities*. The MIT Press. <https://doi.org/10.7551/mitpress/9248.003.0003>
- Burnard, L. (2012). Du literary and linguistic computing aux digital humanities : retour sur 40 ans de relations entre sciences humaines et informatique. In P. Mounier (Éd.), *Read/Write Book 2 : Une introduction aux humanités numériques* (p. 45-58). OpenEdition Press. <https://doi.org/10.4000/books.oep.242>
- Burnard, L. (2015). *Qu'est-ce que la Text Encoding Initiative ?* (M. Burghart, Trad.). OpenEdition Press. <https://doi.org/10.4000/books.oep.1237>
- Cardon, D. (2013). Dans l'esprit du PageRank:Une enquête sur l'algorithme de Google. *Réseaux*, 177(1), 63-95. <https://doi.org/10.3917/res.177.0063>
- Chagué, A. (2022). Intelligence Artificielle et Intelligence Collective : Des Nouveaux Eldorados Pour Rendre Les Textes Patrimoniaux plus Accessibles? [Medium]. In *Museonum*. https://hal.science/hal-03739948_v1
- Collectif. (2018). Pourquoi Zotero? [{Billet}]. In *Le blog Zotero francophone*. <https://doi.org/10.58079/vdwp>
- Commission d'enrichissement de la langue française. (2019). *Journal Officiel Electronique Authentifié N° 0157*. https://www.legifrance.gouv.fr/download/pdf?id=rtw17ZyCloLK1ylXH84-zZnRB4DS_m6KQ1OzvTQ6mNw=
- Coombs, J. H., Renear, A. H., & DeRose, S. J. (1987). Markup Systems and the Future of Scholarly Text Processing. *Communication of the ACM*, 30(11), 933-947. <https://doi.org/10.1145/32206.32209>
- Dacos, M. (2011). Manifeste des Digital humanities [{Billet}]. In *THATCamp Paris*. <https://doi.org/10.58079/uo27>
- Dacos, M., & Mounier, P. (2015). *Humanités Numériques* [Research {{Report}}]. Institut français. <https://hal.science/hal-01228945>
- Daniel, J. (2022). Ode à Zotero, une décennie d'usage de Zotero [{Billet}]. In *Isidore & Ganesh*. <https://doi.org/10.58079/pw5v>
- Debouy, E. (2025). *Vade-Mecum Informatique Pour Lettres et Sciences Humaines* (P. universitaires de Rennes, Éd.).
- Drucker, J. (2011). Humanities Approaches to Graphical Display. *Digital Humanities Quarterly*, 005(1). <https://dhq.digitalhumanities.org/vol/5/1/000091/000091.html>

- Fauchié, A., & Audin, Y. (2023). The Importance of Single Source Publishing in Scientific Publishing. *Digital Studies / Le Champ Numérique, Special DSCN Collection #9*. <https://doi.org/10.16995/dscn.9655>
- France Inter. (s. d.). *Le Français qui a vu naître Google*. <https://www.radiofrance.fr/franceinter/podcasts/le-code-a-change/le-francais-qui-a-vu-naître-google-4402005>
- Fredriksson, S., & Sauret, N. (2019). Écrire les communs. Au-devant de l'irréversible. *Sens public*. <http://sens-public.org/dossiers/1383/>
- Huma-Num. (2021). ISIDORE a 10 ans ! [{Billet}]. In *Le blog d'Huma-Num et des Consortiums-HN*. <https://doi.org/10.58079/pr1t>
- Jahjah, M. (2017). "Annoter Le Monde et Améliorer l'humanité" : Les Imaginaires Matériels d'un Logiciel d'annotation Du Web. In *Ecrilecture Augmentée Dans Les Communautés Scientifiques - Humanités Numériques et Construction Des Savoirs*. <https://hal.science/hal-01638125>
- Just What Is TeX? (2023). In *TeX Users Group*. <http://ftp.tug.org/whatis.html>
- Lamport, L. (1994). *La TeX: A Document Preparation System*. Addison-Wesley Professional.
- Larsonneur, C. (2008). *La recherche Internet en lettres et langues*. Editions OPHRYS.
- Lozano, V. (2013). *Tout ce que vous avez toujours voulu savoir sur LaTeX sans jamais oser le demander ou comment utiliser LaTeX quand on n'y connaît goutte* (Nouvelle éd.). Framasoft.
- Mille, A. (2014). Chapitre 2. D'Internet au web. In M. Vitali-Rosati & M. E. Sinatra (Éds.), *Pratiques de l'édition numérique* (p. 31-48). Presses de l'Université de Montréal. <https://doi.org/10.4000/books.pum.315>
- Ochigame, R., & Ye, K. (2021). Search Atlas: Visualizing Divergent Search Results Across Geopolitical Borders. *Proceedings of the 2021 ACM Designing Interactive Systems Conference, 1970-1983*. <https://doi.org/10.1145/3461778.3462032>
- Perret, A. (2022a). À quoi sert une vue graphe ? In *arthurperret.fr*. <https://www.arthurperret.fr/blog/2022-02-17-analyser-synthétiser-visualiser.html>
- Perret, A. (2022b). Analyser, synthétiser, visualiser : le triptyque fiche, lien, graphe. In *arthurperret.fr*. <https://www.arthurperret.fr/blog/2022-02-13-a-quoi-sert-une-vue-graphe.html>
- Reboul, M., & Gefen, A. (2019). Mesures et savoirs : Quelles méthodes pour l'histoire culturelle à l'heure du big data ? *Semiotica, 2019*(230), 97. <https://doi.org/10.1515/sem-2018-0103>
- Robichaud, L., Burgess, J., & Linteau, P.-A. (2019). Prendre Le Tournant Spatial En Histoire: Le Laboratoire d'histoire et de Patrimoine de Montréal et Le Système de Cartographie de l'histoire de Montréal. In *Histoire et Patrimoine. Pistes de Recherche et de Mise En Valeur* (p. 213-236). Presses de l'Université Laval. <https://www.pulaval.com/libreacces/9782763743295.pdf>
- Schmidt, B. M. (2016). Do Digital Humanists Need to Understand Algorithms? *Debates in the Digital Humanities 2016*, 546-555. <https://doi.org/10.5749/j.ctt1cn6tb.51>
- Schrijver, É. (2017). Culture hacker et peur du WYSIWYG. In *Back Office*. <http://www.revue-backoffice.com/numeros/01-faire-avec/eric-schrijver-culture-hacker-peur-wysiwyg>
- Sinatra, M. E., & Vitali-Rosati, M. (2014). Histoire des humanités numériques. In *Pratiques de l'édition numérique*. Les Ateliers de [sens public]. <https://www.parcoursnumeriques-pum.ca/1-pratiques/chapitre-3.html>
- Smits, T., & Wevers, M. (2023). A Multimodal Turn in Digital Humanities. Using Contrastive Machine Learning Models to Explore, Enrich, and Analyze Digital Visual Historical Collections. *Digital Scholarship in the Humanities, 38*(3), 1267-1280. <https://doi.org/10.1093/llc/fqad008>
- Turbiau, A. (2023). Rédaction : gérer les différences de versions avec Git [{Billet}]. In *Littératures engagées*.

- <https://doi.org/10.58079/oe11>
- Underwood, T. (2018). Why an Age of Machine Learning Needs the Humanities. In *Public Books*. <https://www.publicbooks.org/why-an-age-of-machine-learning-needs-the-humanities/>
- Vitali Rosati, M., Sauret, N., Fauchié, A., & Mellet, M. (2020). Écrire les SHS en environnement numérique. L'éditeur de texte Stylo. *Revue Intelligibilité du numérique*, 1. <https://intelligibilite-numerique.numerev.com/numeros/n-1-2020/18-ecrire-les-shs-en-environnement-numerique-l-editeur-de-texte-stylo>
- Vitali-Rosati, M. (2018). Les Chercheurs En SHS Savent-Ils Écrire ? In *The Conversation*. <http://theconversation.com/les-chercheurs-en-shs-savent-ils-ecrire-93024>

Contenu

Plan du cours	2
Description du cours	2
Objectifs et contenu du cours	2
Organisation du cours	3
Modalités d'évaluation	4
Examen de mi-session	4
Travail bonus : <i>Apprenez-moi quelque chose</i>	4
Examen final	5
Calendrier des séances et des lectures	6
Ressources utilisées dans le cadre de ce cours	6
Plagiat et fraude	6
Crédits	6
License	6
Calendrier des séances et des lectures	7
[Séance 1 : 02/09] Introduction au cours et aux Humanités Numériques	7
[Séance 2 : 09/09] COMPRENDRE son environnement	8
[Séance 3 : 16/09] CHERCHER : corpus et ressources numériques	9
[Séance 4 : 23/09] ORGANISER & BIBLIOGRAPHIER : métadonnées et formats	10
[30/09]	10
[Séance 5 : 07/10] REPRÉSENTER le texte (I) : formats légers	11
[Séance 6 : 14/10] REPRÉSENTER le texte (II) : formats riches et normés	12
[21/10]	12
[Séance 7 : 28/10] EXAMEN	13
[Séance 8 : 04/11] ANNOTER : des <i>marginalia</i> à l'annotation numérique	14
[Séance 9 : 11/11] ÉCRIRE & PUBLIER	15
[Séance 10 : 18/11] NETTOYER & EXPLOITER les données textuelles	16
[Séance 11 : 25/11] ANALYSER : textométrie, stylométrie et TAL	17
[Séance 12 : 02/12] VISUALISER : vers une autre lecture	18
[Séance 13 : 09/12] IA	19
[Séance 14 : 16/12]	20
Introduction au cours	21
Bienvenue !	21
Tour de table	21

Déroulement du cours	21
Objectifs du cours	21
Présentation des séances	22
Modalités d'évaluation	22
[Séance 1 : 02/09] Introduction aux Humanités Numériques	23
Définitions	23
Humanités...	23
...Numérique(s)	24
Les trois moments des Humanités Numériques	24
env. 1970 : <i>Literary and Linguistic Computing</i>	24
env. 1980 : <i>L'humanities computing</i>	24
env. 1990 : <i>Digital Humanities</i>	24
Quelques repères historiques	25
Les prémisses mécaniques	25
Les concepts fondateurs	27
Vers l'ordinateur moderne	28
L'informatique et les sciences humaines	28
Démocratisation et réseau	28
Vers les humanités numériques contemporaines	30
[Séance 2 : 09/09] COMPRENDRE son environnement	31
Retour sur la lecture	31
Terminal, ligne de commande	31
La ligne de commande	31
Exemple	32
La ligne de commande, plus rapide et plus sûre ?	32
Testons	32
Où sont les fichiers sur un ordinateur ?	32
Arborescence	33
Nommer ses fichiers	33
Chemins relatifs et chemins absolus	34
Les premières commandes	34
Où suis-je ?	34
Déplacer un fichier	35
Naviguer dans ses fichiers	35
Lister des fichiers	35
Créer un dossier	35
Créer un fichier	35
Supprimer un fichier	36
Afficher le contenu d'un fichier	36
Comprendre une commande	36
Quelques réflexes	36

Exercice	36
Quelques commandes avancées	36
Bash	37
Exercices	37
WYSIWYM >< WYSIWYG ?	39
WYSIWYM	40
Mettre en forme =/= structurer	40
VSCodium : installation et manipulations	40
C'est quoi Codium ?	40
Extensions	41
[Séance 3 : 16/09] CHERCHER: corpus et ressources numériques	42
Retour sur les lectures et écoutes	42
Une histoire d'internet et du Web	42
Origines	42
Principes : TCP/IP	42
Expérimentation	43
Le Web	43
Exercice	44
Les navigateurs et moteurs de recherche	44
Le Web, au commencement étaient les annuaires	44
Explorer, indexer, classer	45
Quelques moteurs de recherche	45
Google et le PageRank	47
Les évolutions des moteurs de recherche	47
Vade-Mecum d'une recherche efficace	48
Exercice	50
Une alternative aux moteurs de recherche : les flux RSS	50
[Séance 4 : 23/09] ORGANISER & BIBLIOGRAPHIER: métadonnées et formats	51
Retour sur les lectures	51
Les métadonnées	51
Pourquoi les métadonnées ?	51
Les différents formats textuels	51
Les formats : un peu d'étymologie	51
Vers une définition du format	52
Format et édition savante	52
Format et extension	52
Formats libres, formats propriétaires	53
Le format n'est pas neutre !	53
Zoom sur le libre	53
Qu'est-ce qu'une bibliographie ?	54
L'importance de structurer ses références bibliographiques	54

Style de citation vs style bibliographique	54
Exemple	55
Comment créer une bibliographie ?	55
Zotero	55
Qu'est-ce que Zotero ?	55
Fonctionnalités	56
Un logiciel libre	56
Utilisation générale	56
Fonctionnalités proposées avec un compte	56
À quoi sert un groupe ?	57
Installation	57
De nombreuses bases de données compatibles	57
Description d'une utilisation classique	57
Quelques points importants	58
Collecter	58
Créer	58
Organiser	58
Exercice : Zotero	59
Exercice 1	59
Exercice 2 : Traitement de texte	59
BibTeX	59
Better BibTeX	60
Installez Better BibTeX	60
Exercices	60
[Séance 5 : 07/10] REPRÉSENTER le texte (I): formats légers	61
Retour sur les lectures	61
Objectifs de la séance	61
Représenter le texte ?	61
Ce poème est-il le même ?	62
Le texte et l'ordinateur	62
Encodage, format, balisage	62
Une courte histoire du balisage	63
SGML, HTML et XML	63
Les formats légers	63
Définition	64
Exemples	64
Une philosophie éditoriale	64
Le format texte	64
.md	65
Markdown : exercice	65
Stylo : mise en pratique	65
C'est quoi ?	65

Discussion	66
Exercice 1	66
Exercice 2	66
HTML	67
Exercice HTML	67
Ex. 1 : Structure minimale & typographie	67
Ex. 2 : Liens, images, listes	67
Ex. 3 : Balises sémantiques & mini défi Markdown→HTML	68
[Séance 6 : 14/10] REPRÉSENTER le texte (II): formats riches et normés	69
Retour sur les lectures	69
De quoi parle-t-on ?	69
(La)TeX : un peu de contexte	69
TeX ou LaTeX ?	70
(La)TeX, concrètement	70
Esprit LaTeX	71
Pourquoi LaTeX ?	71
LaTeX : Les indispensables	71
Classe du document	72
Préambule	72
Corps du document	72
Quelques principes	73
Comparaison Markdown, HTML, LaTeX	73
Un exemple de document LaTeX minimal	73
Exercice : Prise en main d'OverLeaf	74
LaTeX : Pour aller plus loin	75
Et après Overleaf ?	76
Windows	76
Mac OS	76
XML-TEI	76
XML	76
TEI	77
Exercice : Écrire en TEI-XML (avec Leaf-Writer)	80
Extra : Quelques outils des Humanités numériques	80
Hypothes.is	80
Stylo	81
Zotero	81
Ngram Viewer	82
Voyant Tools	82
Gephi	83
Qu'est-ce qu'une étude de cas ?	84

[Séance 9 : 11/11] ÉCRIRE & PUBLIER	85
Retour sur les lectures	85
Le Single Source Publishing	85
Une seule source	85
Modéliser les sorties	86
Quelques exemples	86
Une approche modulaire	86
Quelques exemples	86
Un exemple d'outil : Quarto	86
Note sur Pandoc	87
Manipuler quarto	87
La gestion de versions	88
Décentraliser pour mieux travailler	88
Les différents systèmes de contrôle de version	88
Introduction à Git	89
Versionnement	89
Exemple (Debouy, 2025, p. 82-83)	89
Quelques principes	90
Une illustration	91
Quelques termes	92
En somme	93
Manipulation : Git et GitHub.	93
Les commandes usuelles	93
Installation de Git	93
Clés SSH	94
Prise en main de Git	94
Tuto	97
Résumé des commandes	97
Exercice	97
Exercice : résoudre un conflit	98
Références	100