

# **HNU6051-A24**

**Humanités numériques : formats d'écriture**

[\*\*Mathilde Verstraete\*\*](#)

Automne 2024

# Contenu

<b>Bienvenue</b>	<b>8</b>
Support de cours . . . . .	8
Crédits . . . . .	8
License . . . . .	9
<b>Plan de cours</b>	<b>10</b>
Informations pratiques . . . . .	10
Description et objectifs de l'atelier . . . . .	10
Support de cours . . . . .	10
Organisation des séances . . . . .	11
Séance 1 (4 octobre) : . . . . .	11
Séance 2 (11 octobre) : . . . . .	11
Séance 3 (18 octobre) : . . . . .	12
Évaluation . . . . .	12
Exercices durant les séances : 20% . . . . .	12
Exercices hebdomadaires : 30% . . . . .	12
Travail final : 50% . . . . .	12
Références et ressources . . . . .	13
Intégrité . . . . .	13
Crédits . . . . .	13
License . . . . .	13
<b>I session 1</b>	<b>14</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Lecture du plan de cours . . . . .	15
1.2 Déroulement de l'atelier . . . . .	15
1.3 Tour de table . . . . .	15
1.4 En ce qui me concerne . . . . .	15
1.5 Me contacter . . . . .	16
<b>2 Les formats : théorie</b>	<b>17</b>
2.1 Les formats : un peu d'étymologie . . . . .	17
2.2 Vers une définition du format . . . . .	17

2.3 Format et édition savante . . . . .	17
2.4 Format et extension . . . . .	18
2.5 Formats libres, formats propriétaires . . . . .	18
2.6 Le format n'est pas neutre! . . . . .	18
2.7 Critères d'un format . . . . .	19
2.7.1 Pertinence . . . . .	19
2.7.2 Économie . . . . .	19
2.7.3 Efficacité . . . . .	19
2.7.4 Sécurité . . . . .	19
2.8 Format, logiciel, extension . . . . .	19
2.9 Structuration de l'information . . . . .	20
2.10 Formats de balisage . . . . .	20
2.10.1 Texte brut, texte enrichi . . . . .	20
2.10.2 WYSIWYM vs WYSIWYG . . . . .	20
2.10.3 WYSIWYM . . . . .	21
2.11 Mettre en forme =/= structurer . . . . .	21
2.12 Markdown . . . . .	21
2.13 LaTeX . . . . .	22
2.14 CSV . . . . .	22
2.15 Questions? . . . . .	23
<b>3 Bonnes pratiques d'édition</b>	<b>24</b>
3.1 Les fichiers dans votre ordinateur . . . . .	24
3.1.1 Organisation des fichiers . . . . .	24
3.1.2 Nommage de fichier . . . . .	25
3.2 De l'importance de la documentation . . . . .	25
<b>4 Le texte brut</b>	<b>26</b>
4.1 <i>Plain text / Plein texte / Texte brut</i> . . . . .	26
4.2 Texte brut : introduction . . . . .	26
4.2.1 Simplicité . . . . .	26
4.2.2 Absence de style . . . . .	26
4.2.3 Universalité . . . . .	26
4.2.4 Interopérabilité . . . . .	27
4.2.5 Poids . . . . .	27
4.3 Pourquoi utiliser le texte brut? . . . . .	27
4.4 Un exemple : The Plain Text Project . . . . .	27
<b>5 Exercice markdown</b>	<b>28</b>
5.1 Markdown : langage de balisage léger . . . . .	28
5.2 Tutoriel (à faire en classe) . . . . .	28
5.3 Rendu pour le 10 octobre . . . . .	28

<b>6 Chaînes éditoriales &amp; Stylo</b>	<b>29</b>
6.1 Chaînes éditoriales . . . . .	29
6.2 Définition . . . . .	29
6.3 Chaînes éditoriales variées . . . . .	29
6.4 Stylo : un projet de recherche & un éditeur de texte . . . . .	31
6.4.1 Démonstration . . . . .	32
6.5 Stylo en mode avancé . . . . .	32
6.6 Rendu pour le 17 octobre . . . . .	32
<b>7 Zotero</b>	<b>33</b>
7.1 L'importance de structurer ses références bibliographiques . . . . .	33
7.2 Zotero : introduction . . . . .	33
7.2.1 Qu'est-ce que Zotero ? . . . . .	33
7.2.2 Fonctionnalités . . . . .	33
7.2.3 Un logiciel libre . . . . .	34
7.3 Utilisation de Zotero . . . . .	34
7.3.1 Utilisation générale . . . . .	34
7.3.2 Fonctionnalités proposées avec un compte . . . . .	34
7.3.3 Installation . . . . .	34
7.3.4 Exemple à partir du catalogue de la Bibliothèque universitaire de l'UdeM	35
7.3.5 De nombreuses bases de données compatibles . . . . .	35
7.3.6 Description d'une utilisation classique . . . . .	35
7.3.7 Quelques points importants . . . . .	35
7.3.8 Collecter . . . . .	36
7.3.9 Créer . . . . .	36
7.4 Organiser . . . . .	36
7.4.1 Collections . . . . .	36
7.4.2 Tags et marqueurs . . . . .	36
7.4.3 Champ « note » . . . . .	37
7.5 Exercice . . . . .	37
7.6 Zotero avancé . . . . .	37
7.7 Crédit de bibliographies . . . . .	37
7.7.1 Style de citation vs style bibliographique . . . . .	37
7.7.2 Comment créer une bibliographie ? . . . . .	38
7.7.3 Citer et créer une bibliographie directement dans un traitement de texte	39
7.8 Crédit et gestion de groupes . . . . .	39
7.8.1 À quoi sert un groupe ? . . . . .	39
7.9 BibTeX . . . . .	39
7.10 Better BibTeX . . . . .	40
7.10.1 Installez Better BibTeX . . . . .	40
7.10.2 Exercices . . . . .	40
7.11 Questions ? . . . . .	40

<b>II session 2</b>	<b>41</b>
<b>8 Le <i>single source publishing</i></b>	<b>42</b>
8.1 Kézako . . . . .	42
8.2 Chaîne éditoriale . . . . .	42
8.2.1 Les différentes étapes de la chaîne éditoriale type . . . . .	43
8.2.2 Et le numérique ? . . . . .	44
8.2.3 Rappel : plusieurs approches de la chaîne éditoriale . . . . .	44
8.3 Les chaînes d'édition modulaires . . . . .	46
8.4 Le <i>single source publishing</i> ou la publication à source unique . . . . .	46
8.4.1 Quelques termes . . . . .	46
8.4.2 Définition . . . . .	47
8.4.3 Ex. : un livre en <i>SSP</i> . . . . .	47
8.4.4 Ex. : un site en <i>SSP</i> . . . . .	48
8.4.5 Intérêts et limites . . . . .	48
8.5 Quelques exemples . . . . .	48
<b>9 Installation de <i>Codium</i></b>	<b>49</b>
9.1 C'est quoi <i>Codium</i> ? . . . . .	49
9.2 Installation . . . . .	49
9.3 Extensions . . . . .	49
<b>10 Utiliser son terminal</b>	<b>50</b>
10.1 Terminal, console, shell... ? . . . . .	50
10.1.1 Le <i>shell</i> . . . . .	50
10.1.2 Terminal . . . . .	51
10.1.3 Émulateur de terminal . . . . .	51
10.2 les CLI ( <i>command-line interfaces</i> ) . . . . .	51
10.3 Intérêts . . . . .	51
10.4 Quelques commandes . . . . .	52
10.4.1 Chemin absolu / chemin relatif . . . . .	52
10.4.2 Quelques bons réflexes . . . . .	52
10.4.3 Gérer des fichiers ou des dossiers . . . . .	52
10.4.4 Déplacer un fichier . . . . .	53
10.5 Rendu pour le 18 octobre . . . . .	53
<b>11 LaTeX</b>	<b>54</b>
11.1 LaTeX : introduction . . . . .	54
11.2 TeX, LaTeX, et al. . . . .	54
11.3 Structure générale . . . . .	54
11.4 Modules supplémentaires, <i>paquets</i> . . . . .	55
11.5 Discussion : avantages, inconvénients . . . . .	55
11.6 Tuto . . . . .	55

<b>III session 3</b>	<b>56</b>
<b>12 Des caractères à l'écran – encodage numérique</b>	<b>57</b>
12.1 Encodage numérique . . . . .	57
12.2 Système de numération en base 2 . . . . .	57
12.3 Mémoire . . . . .	59
12.4 0 ou 1 ? . . . . .	59
12.5 Normes d'encodage . . . . .	59
12.6.1 ASCII . . . . .	60
12.6.2 ISO 8859-1 . . . . .	61
12.6.3 Unicode . . . . .	61
12.6.4 UTF-8 . . . . .	62
12.7 Mauvais encodage/décodage . . . . .	62
12.8 Une norme ? . . . . .	63
<b>13 Convertisseurs : le cas de <i>pandoc</i></b>	<b>64</b>
13.1 Pandoc . . . . .	64
13.2 Fonctionnement (en ligne de commande) . . . . .	64
13.3 Installation (en classe) . . . . .	65
13.4 Quelques options de la commande Pandoc . . . . .	65
13.5 Exemples supplémentaires . . . . .	65
13.6 Métadonnées (exercice) . . . . .	66
13.7 Export d'une bibliographie . . . . .	66
13.8 Les <i>templates</i> . . . . .	67
13.9 Personnaliser son <i>template</i> (exercice) . . . . .	67
13.10 Questions ? . . . . .	68
<b>14 Quarto</b>	<b>69</b>
14.1 Quarto . . . . .	69
14.2 Fonctionnement . . . . .	69
14.3 Historique . . . . .	69
14.4 Au travail . . . . .	70
<b>15 Git et GitHub</b>	<b>71</b>
15.1 Versionnement . . . . .	71
15.2 Git vs Github . . . . .	71
15.2.1 Git . . . . .	71
15.2.2 GitHub . . . . .	72
15.2.3 Fonctionnement . . . . .	72
15.3 Quelques commandes de base . . . . .	72
15.3.1 Quelques commandes un peu plus complexes . . . . .	73
15.4 Git chatons . . . . .	73

15.5 Exercice - créer son premier répertoire . . . . .	74
15.6 Installation de Git . . . . .	75
<b>16 YAML</b>	<b>76</b>
16.1 Définition . . . . .	76
16.2 Syntaxe de base . . . . .	76
16.3 Quarto et yaml . . . . .	76
16.4 Utilisation avancée . . . . .	77
16.5 Exercice . . . . .	77
<b>Références</b>	<b>78</b>

# Bienvenue

Bienvenue sur le support du cours **HNU6051 – Humanités numériques : formats d’écriture**. Celui-ci est destiné aux étudiantes et étudiants qui suivent l’atelier à l’automne 2024. Les séances se tiendront les vendredi 4, 11, 18 octobre, de 9h30 à 11h30 puis de 12h30 à 15h30 au Pavillon Lionel-Groulx, salle C-3115.

L’objectif de cet atelier est de découvrir les différents formats d’écriture et de prendre en main le fonctionnement de différentes chaînes éditoriales de production de contenu.

Aujourd’hui de nombreux outils ou applications de rédaction, d’édition et de publication de contenus textuels sont accessibles au grand public sur le Web, mais sans nécessairement prendre en compte les besoins et les pratiques d’écriture des communautés de recherche. La question des formats se place au cœur de la structuration du texte numérique et il est important, pour qui veut produire des connaissances organisées, pérennes et interopérables, de comprendre les enjeux culturels, techniques et politiques des formats.

Ces trois séances entendent donner aux étudiantes et étudiants une connaissance théorique et pratique sur les différents formats d’écriture, leur donner les clés pour mettre en place de bonnes pratiques éditoriales et les former à quelques outils propices à la rédaction scientifique.

## Support de cours

Le présent support est destiné aux étudiantes et étudiants de l’Université de Montréal suivant l’atelier [HNU6051, Humanités numériques : formats d’écriture](#), au trimestre d’Automne 2024.

## 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](#).

Le support du cours a été créé avec [quarto](#), les slides suivent un modèle développé par [Alison Presmanes Hill](#).

## **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.

# Plan de cours

## Informations pratiques

- Sigle du cours : HNU6051
- Intitulé du cours : Humanités numériques : formats d'écriture
- Chargée de cours : [Mathilde Verstraete](#)
- Horaire : trois vendredi : 4, 11, 18 octobre 2024, de 9h30 à 11h30 puis de 12h30 à 15h30
- Lieu : Pavillon Lionel-Groulx, salle C-3115.

## Description et objectifs de l'atelier

La question des formats se place au cœur de la structuration du texte numérique et il est important, pour qui veut produire des connaissances organisées, pérennes et interopérables, de comprendre les enjeux culturels, techniques et politiques des formats. Aujourd’hui de nombreux outils ou applications de rédaction, d'édition et de publication de contenus textuels sont accessibles au grand public sur le Web. Ces outils ne prennent pas nécessairement en compte les besoins et les pratiques d'écriture des communautés de recherche.

Ces trois séances entendent donner aux étudiantes et étudiants une connaissance théorique et pratique sur les différents formats d'écriture, leur donner les clés pour mettre en place de bonnes pratiques éditoriales et les former à quelques outils propices à la rédaction scientifique.

L'atelier n'a pas seulement pour but de former à des outils spécifiques des Humanités numériques mais vise à offrir aux étudiant · e · s les connaissances théoriques et pratiques fondamentales pour leur permettre d'explorer les potentialités du numérique, de constituer une réflexion critique sur les outils numériques et de développer leurs propres pratiques d'écriture et de publication.

## Support de cours

Le support de cours est multiformat. Un site Web est à votre disposition à l'adresse suivante : <https://mathildevrst.github.io/hnu6051-a24/>. Vous pourrez y télécharger le PDF et afficher les slides de chaque séance.

## Organisation des séances

Nous alternerons entre pratique théorie et pratique. Il est conseillé de venir en cours avec son ordinateur personnel.

### ! Important

Le programme des séances est sujet à de légers changements en fonction du niveau et de l'avancement du groupe.

### Séance 1 (4 octobre) :

Horaire	Matière	Type
09h30-10h00	<a href="#">Présentation &amp; Introduction de l'atelier</a>	—
10h00-10h50	Édition savante, formats de fichiers	Théorie
10h50-11h10	<a href="#">Bonnes pratiques de l'édition</a>	Théorie
11h10-11h30	The Plain Text Project	Théorie et exercice
11h30-12h30	Pause	—
12h30-13h00	<a href="#">Markdown</a>	Théorie et exercice
13h00-14h15	Chaînes éditoriales, Stylo	Théorie et pratique
14h15-15h15	<a href="#">Références structurées, Zotero</a>	Théorie et exercice
15h15-15h30	Présentation du rendu pour la séance 2 (markdown)	Rendu

### Séance 2 (11 octobre) :

Horaire	Matière	Type
09h30-10h00	Retour sur le rendu	Rendu
10h00-11h15	<a href="#">Single Source Publishing</a>	Théorie
11h15-11h30	<a href="#">Installation de Codium</a>	Pratique
11h30-12h30	Pause	—
12h30-13h15	<a href="#">Utilisation du terminal</a>	Pratique
13h15-14h00	<a href="#">Zotero et Bibtex</a>	Pratique
14h00-14h45	<a href="#">Stylo en mode avancé</a>	Exercice
14h45-15h15	<a href="#">Latex</a>	Pratique
15h15-15h30	Présentation du rendu pour la séance 3 (stylo avancé ; recherches terminal)	Rendu

### Séance 3 (18 octobre) :

Horaire	Matière	Type
09h30-10h00	Retour sur le rendu	Rendu
10h00-10h45	Les caractères à l’écran	Théorie
10h45-11h30	Pandoc	Pratique
11h30-12h30	Pause	—
12h30-13h00	<a href="#">Yaml</a>	Théorie et exercices
13h00-14h00	<a href="#">Git et Github</a>	Théorie
14h00-15h15	<a href="#">Quarto</a>	Pratique
15h15-15h30	Présentation du rendu final (création multi-format avec quarto)	Rendu

## Évaluation

### Exercices durant les séances : 20%

Durant les séances de l’atelier, plusieurs exercices seront demandés à l’étudiant · e pour valider au fur et à mesure les acquis techniques. Cette évaluation se fera sur un mode continu.

### Exercices hebdomadaires : 30%

Chaque semaine, en préparation de la séance prochaine, un exercice sera soumis à l’étudiant · e et sera discuté par la suite en classe :

- pour le cours du 11 octobre (soit, la veille) : [Markdown](#) (10%)
- pour le cours du 18 octobre (soit, la veille) : [Stylo en mode avancé](#) (10%) ; [Apprenez-moi quelque chose quant au terminal](#) (10%)

### Travail final : 50%

Le travail final consiste en la création d’un *book* multiformat avec le système de publication [Quarto](#). Le contenu du travail est libre. Je vous invite, dans la mesure du possible, à réutiliser un rendu d’un autre cours.

Quelques critères sont à remplir :

- Le livre doit être présenté en deux formats au moins ;
- Le livre doit comporter au moins 4 *chapitres* ;
- Le thème doit avoir été – au moins légèrement – modifié ;

- Le fichier `_quarto.yml` doit avoir été personnalisé ;
- Une publication accessible en ligne (via *github pages*) sera particulièrement appréciée ;
- Date de remise : vendredi 8 novembre.

## Références et ressources

Une liste de références et de ressources variées sera mise à jour tout au long du cours. Elle est disponible [en cliquant ici](#).

## Intégrité

L’Université de Montréal s’est dotée de deux règlements disciplinaires sur la fraude et le plagiat. L’une s’adresse aux étudiant · e · s de 1<sup>er</sup> cycle, la seconde aux étudiant · e · s des cycles supérieurs : <https://integrite.umontreal.ca/boite-a-outils/les-reglements>.

## 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*.

Le support du cours a été créé avec [quarto](#), les slides suivent un modèle développé par [Alison Presmanes Hill](#).

## 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.

**Première partie**

**session 1**

# 1 Introduction

## 1.1 Lecture du [plan de cours](#)

## 1.2 Déroulement de l'atelier

- Un peu de théorie
- Beaucoup de pratique : on met les mains à la pâte, on teste des outils, on produit, on fabrique !

Vous n'êtes pas un groupe homogène donc vous n'irez pas toutes et tous à la même vitesse, pas d'inquiétude !

## 1.3 Tour de table

Quelques informations pour savoir qui vous êtes, quel est l'objectif de cet atelier et comment les séances vont être organisées. 5 questions :

- Quel est votre nom ?
- Où étiez-vous l'année dernière ?
- Quel cursus suivez-vous cette année ?
- Pourquoi suivez-vous cet atelier ?
- Qu'est-ce qui vous vient instinctivement à l'esprit quand je vous dis format ?

## 1.4 En ce qui me concerne

En ce qui me concerne, je m'appelle Mathilde Verstrate. Je suis doctorante au Département de littératures et de langues du monde à l'Université de Montréal, dans l'option humanités numériques. J'ai une formation en lettres classiques – que j'ai suivie en Belgique, à l'Université Catholique de Louvain. Je suis passée du côté obscur de la force en rejoignant la [Chaire de recherche du Canada sur les écritures numériques](#) (dirigée par Marcello Vitali-Rosati). C'est à partir de ce moment là que j'ai commencé à sérieusement m'intéresser et à me spécialiser dans les humanités numériques. C'est la première fois que j'enseigne ce cours – j'enseignais un cours de latin les années précédentes.

## 1.5 Me contacter

N'hésitez pas à me contacter. Veillez à vous y prendre à l'avance si vous avez des questions concernant les rendus.

[mathilde.verstraete@umontreal.ca](mailto:mathilde.verstraete@umontreal.ca)

## 2 Les formats : théorie

### 2.1 Les formats : un peu d'étymologie

- Format, fromage ? ;
- 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**

### 2.2 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](#))

### 2.3 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'expliquer la fonction de chaque élément du document (titre, citation, métadonnées, références bibliographiques,...)

## 2.4 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.

## 2.5 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

## 2.6 Le format n'est pas neutre !

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

## 2.7 Critères d'un format

Le format répond à plusieurs critères : pertinence, économie, efficacité, sécurité.

### 2.7.1 Pertinence

Certains formats sont plus pertinents que d'autres, dépendant de la discipline, de l'utilisation, de la langue, du contexte,...

Ex. : .pdf a été conçu spécifiquement pour la mise en page de documents prévus pour l'impression

### 2.7.2 Économie

Quel modèle soutient le format ? Quel poids du document ? Pour quelle possibilité de stockage ?

Ex. : un texte à envoyer rapidement (une note) doit être relativement léger ; pour un texte plus important, on peut prioriser des fonctionnalités plus complexes, plus permissives mais qui alourdissent le fichier.

### 2.7.3 Efficacité

Quelle temporalité pour l'écriture et la finalisation du document ?

### 2.7.4 Sécurité

Le contenu est-il garanti ? Quelles protection pour les données ? Quelle pérennité pour le document ?

## 2.8 Format, logiciel, extension

- un format n'est pas un logiciel : parler du format Word, c'est faux ;
- un format n'est pas une extension : changer l'extension ne change pas le format.

Le format va structurer une information, va lui donner une forme, va délimiter son utilisabilité, les manipulations qui vont être possibles ou non.

## 2.9 Structuration de l'information

Nous devons faire la distinction entre :

- le corps du document,
- les métadonnées,
- le contenu additionnel, comme la bibliographie.

## 2.10 Formats de balisage

- un format de balisage : des balises pour structurer et qualifier le texte (.md, .html, .xml) ;
- un format de balisage léger : compréhensible par les humains ;
- un format transformable : format source facilement interprétable pour obtenir un format de sortie riche
- un format pérenne : encore utilisable dans 10 ans (car porté par une communauté et des standards)



Tip

Lecture recommandée : (Coombs, Renear, and DeRose 1987).

### 2.10.1 Texte brut, texte enrichi

Les informations de mise en forme sont des annotations qui sont formées selon plusieurs standards ou normes.

- texte brut (ou *plain text*) : texte dépourvu d'information de mise en forme, qui se limite à l'affichage de la traduction des données binaires en caractères alphanumériques et symboles ;
- texte enrichi (ou *rich text*) : texte qui comporte des éléments de mise en forme, qu'ils soient ou non affichés dans l'interface graphique ; Un fichier de texte peut contenir des indications de mise en forme.

Les normes et standards encadrant la manière de former les annotations sont étroitement liées au format du document et au standard qu'il suit.

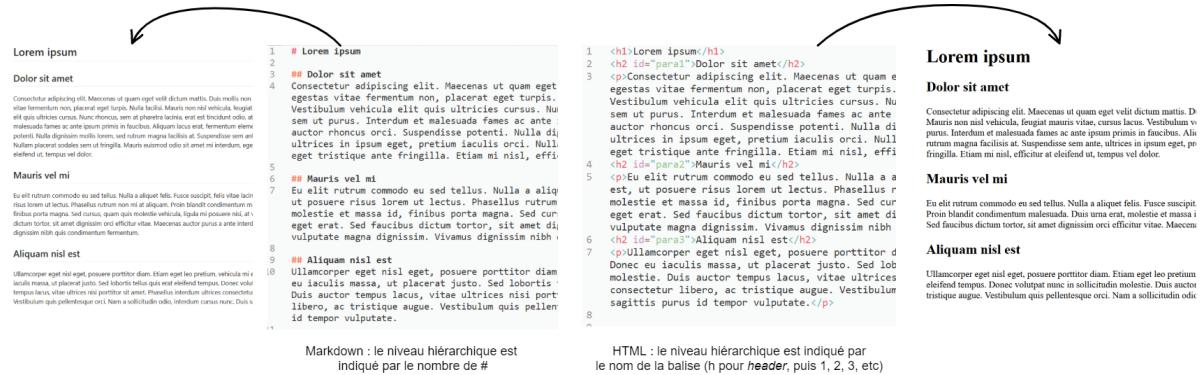
### 2.10.2 WYSIWYM vs WYSIWYG

WYSIWYM	WYSIWYG
<p><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</p>	<p><i>What You See Is What You Get</i> éditeur de texte qui présente un texte enrichi <b>avec</b> sa mise en forme, de manière simultanée</p>

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

### 2.10.3 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).



## 2.11 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

## 2.12 Markdown

- Inventé en 2004 par John Gruber et Aaron Swartz :

- Markdown est un langage de balisage léger qui permet d'encoder facilement la structure et certains éléments de mise en forme du texte sans gêner sa lecture ;
- Il permet de créer des fichiers HTML ou PDF et est beaucoup utilisé sur le Web, notamment sur des plateformes comme Github ;
- Même non compilé, un texte Markdown est facile à lire car les annotation sont légères et limitées ;
- On peut utiliser Markdown ou un système similaire dans Whatsapp et Messenger.

 Tip

Lectures recommandées : (Simpkin 2015 ; Fauchié 2017/2018).

## 2.13 LaTeX

- LaTeX est un langage et un système de composition de documents, bien plus complet que Markdown ;
- Utilisé dans le domaine des sciences car on peut facilement écrire des formules mathématiques ;
- Les balises sémantiques de LaTeX sont plus “invasives”. Exemple : \chapter{Titre de chapitre} ;
- Pour voir la mise en forme, on doit *compiler* le fichier .tex. On obtient généralement un PDF ;
- LaTeX ne sert pas qu'à annoter la structuration logique : il prend en charge la mise en forme du document au sens large.

## 2.14 CSV

CSV (Comma Separated Values) - données tabulaires

- 1 ligne de texte = 1 ligne de tableau ;
- 1 virgule = 1 nouvelle cellule de tableau ;
- un fichier csv peut être ouvert avec Excel.



Article	Quantité	Commentaires
2 Pommes	5	Pour la tarte
3 Bananes	7	Pain aux bananes
4 Riz	"1,5 kg"	Pour demain soir
5 Chocolat	200g	Parce que tout est mieux avec du chocolat!
6 Café	250g	Pour tenir le coup pendant les cours

## 2.15 Questions ?

# 3 Bonnes pratiques d'édition

## 3.1 Les fichiers dans votre ordinateur

### 3.1.1 Organisation des fichiers

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

```
mak@Ubuntu-VM: $ tree -d ~/
/home/mak/
├── Bureau
├── Documents
├── Images
├── Modèles
├── Musique
└── Public
  snap
    ├── firefox
    │   └── 2088
    │       ├── common
    │       │   └── current --> 2088
    │       └── snapd-desktop-integration
    │           ├── 18
    │           │   ├── Bureau
    │           │   ├── Documents
    │           │   ├── Images
    │           │   ├── Modèles
    │           │   ├── Musique
    │           │   ├── Public
    │           │   ├── Téléchargements
    │           │   └── Vidéos
    │           └── 14
    │               ├── Bureau
    │               ├── Documents
    │               ├── Images
    │               ├── Modèles
    │               ├── Musique
    │               ├── Public
    │               ├── Téléchargements
    │               └── Vidéos
    │                   └── common
    │                       └── current --> 14
    └── Téléchargements
        └── Vidéos

34 directories
mak@Ubuntu-VM: $
```

Il faut d'abord comprendre comment est organisé votre ordinateur et où se trouvent vos fichiers. Votre disque dur peut être compris comme une grande boîte dans laquelle se trouvent d'autres boîtes (dossiers). La boîte principale est ce qu'on appelle « racine ». Dans un ordinateur Unix (Mac ou Linux) la racine est /. Dans un ordinateur Windows, la racine est C :/. Dans la racine vous trouvez plusieurs « boîtes ».

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

Il est important que vous connaissiez la structure de l'arborescence de votre ordinateur car c'est ainsi qu'on accède aux fichiers.

### 3.1.2 Nommage de fichier

Quel nom donner à un fichier ? Quelques règles :

- 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 : 2024-10-04-edn6051-bonnes-pratiques.md

Cette question pas si banale que ça. Il existe des règles qui sont fondamentales pour :

- Classer un fichier ;
- Le retrouver facilement ;
- Partager ses fichiers avec d'autres personnes ;
- Éviter des problèmes de comptabilité d'un système à un autre.

## 3.2 De l'importance de la documentation

(Pour toutes pratiques)

- La documentation est votre amie : RTFM ;
- Si vous rencontrez un problème, quelqu'un quelque part a déjà dû le rencontrer : les forums sont vos amis (communautés)

## 4 Le texte brut

### 4.1 Plain text / Plein texte / Texte brut

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

### 4.2 Texte brut : introduction

- 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,..

#### 4.2.1 Simplicité

- Uniquement des caractères alphanumériques, des symboles

#### 4.2.2 Absence de style

- Chaque caractère est traité de manière identique.

#### 4.2.3 Universalité

- Le “plain text” est facilement lisible et compréhensible par n’importe quel éditeur de texte, quel que soit le système d’exploitation ou le logiciel utilisé.

#### 4.2.4 Interopérabilité

- Les fichiers en “plain text” peuvent être utilisés et interprétés par de nombreux systèmes, langages de programmation et applications.

#### 4.2.5 Poids

- Les fichiers .txt sont généralement très légers, car ils n’incluent pas de mises en forme ou d’éléments graphiques. Cela les rend rapides à transférer et à charger.

Faisons le test !

1. Créez un dossier dans votre ordinateur ;
2. Créez un fichier test : `hnu6051-test.txt` ;
3. Ajoutez-y quelques phrases :

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.

4. Sauvez le document ;
5. Faites de même avec un document sur Word. Mettez-en forme le titre, ajoutez de l’italique, du gras... puis sauvez-le ;
6. Comparez la taille des deux documents ;
7. Essayez d’extraire l’archive de votre document .docx (clic droit ; décompresser, extraire vers...)

### 4.3 Pourquoi utiliser le texte brut ?

#### 4.4 Un exemple : [The Plain Text Project](#)

Parcourez le site

Qu’en pensez-vous ?

# 5 Exercice markdown

## 5.1 Markdown : langage de balisage léger

Nous en avons parlé [ici](#)

Markdown	Rendu graphique	Balisage HTML
Un texte _en italique_	Un texte <i>en italique</i>	Un texte <em>en italique</em>
### Titre de niveau 3	<b>Titre de niveau 3</b>	<h3>Titre de niveau 3</h3>
Un [lien] ( <a href="https://www.unlien.org">https://www.unlien.org</a> )	Un <a href="#">lien</a>	Un <a href=" <a href="https://www.unlien.org">https://www.unlien.org</a> ">lien</a>
Un texte **en gras**	Un texte <b>en gras</b>	Un texte <strong>en gras</strong>

## 5.2 Tutoriel (à faire en classe)

<https://www.arthurperret.fr/tutomd/>

## 5.3 Rendu pour le 10 octobre

- Texte : [pont\\_mirabeau.txt](#) ;
- Image : [pont\\_mirabeau.png](#) ;
- Éditeur en ligne : [Dillinger](#)

Faire un encodage en Markdown du poème :

- mettre le titre en niveau 1 ;
- faire de chaque première ligne de strophe un titre de niveau 3 ;
- mettre en italique le paratexte ;
- mettre les noms propres en gras ;
- sauver le document en **.md** ;
- le déposer sur studium / me l'envoyer par mail.

# 6 Chaînes éditoriales & Stylo

## 6.1 Chaînes éditoriales

Encore un peu de théorie avant de prolonger les manipulations.

## 6.2 Définition

- Ensemble des processus, outils et méthodes mis en place pour réaliser une activité d'édition.
- On peut aussi parler de systèmes de publication, de fabriques d'édition ou de chaînes d'édition.

## 6.3 Chaînes éditoriales variées

- *classiques* : traitement de texte + logiciel de publication assistée par ordinateur (essentiellement Adobe InDesign ou Microsoft Publisher) ;
- *savantes* : traitement de texte + logiciel de publication assistée par ordinateur + éditeur XML ;
- *non conventionnelles* : Stylo, Zettlr, Quire, etc.

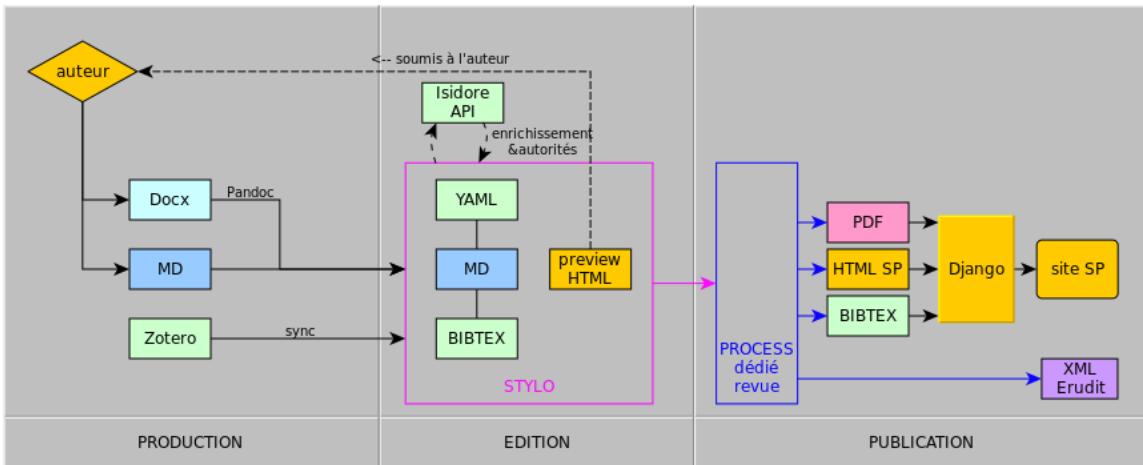


FIGURE 6.1 – Chaîne Stylo et Sens Public

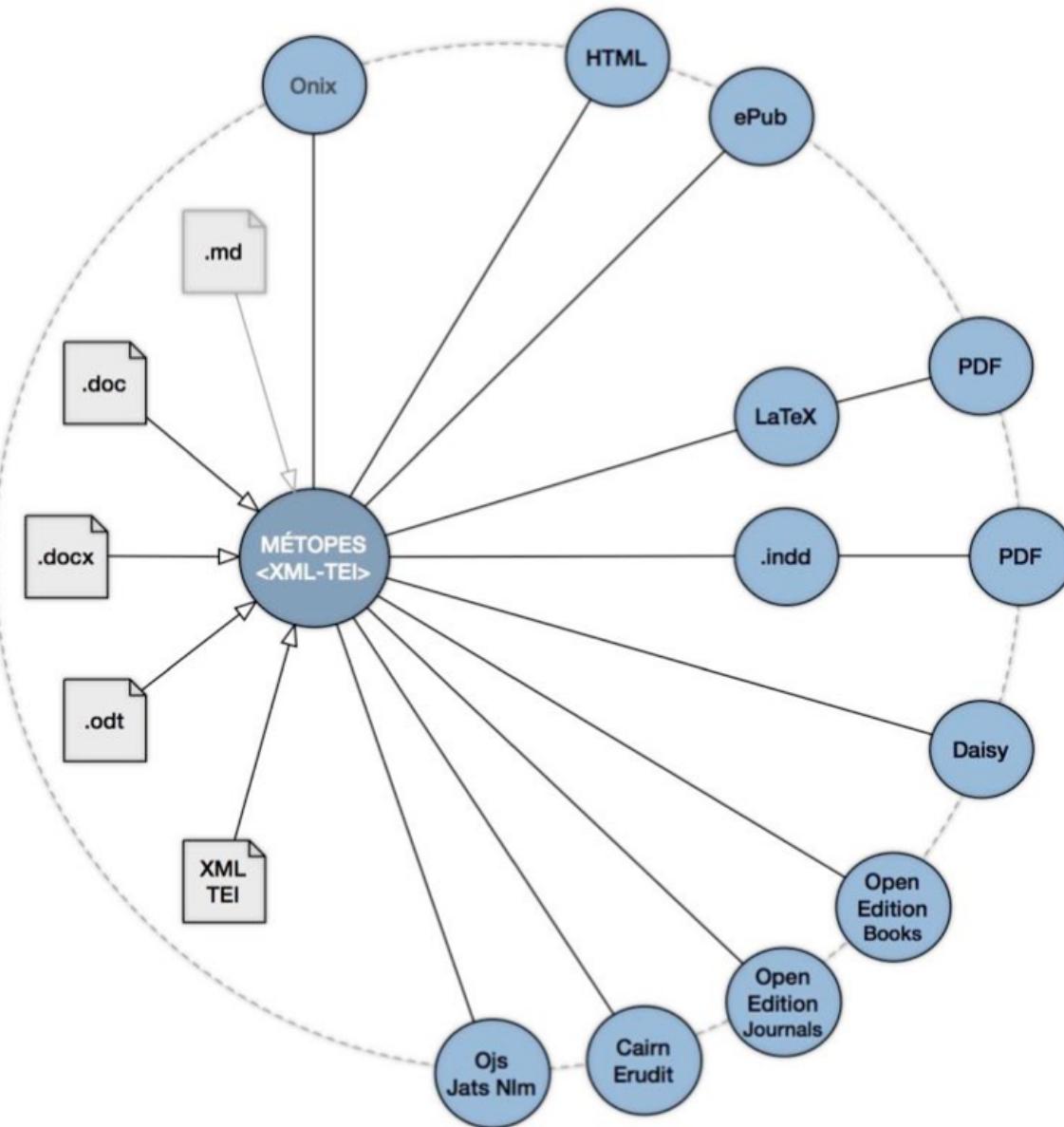


FIGURE 6.2 – Chaîne Métopes

#### 6.4 Stylo : un projet de recherche & un éditeur de texte

- Éditeur de texte sémantique pour la rédaction scientifique ;
- Écrire, structurer, prévisualiser, réviser, exporter ;

- 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).

#### **6.4.1 Démonstration**

1. Crédit à la création d'un compte stylo : <https://stylo.huma-num.fr> ;
2. Reprendre le texte de Catulle [ici](#) ;
3. Ajouter 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](mailto:mathilde.verstraete@umontreal.ca) ;
5. Testez la *preview* et faites différents exports (.html, .pdf, .docx), comparez les résultats obtenus.

Alors *Stylo*, WYSIWYM ou WYSIWYG ?

#### **6.5 Stylo en mode avancé**

Après avoir vu Zotero et BibTex (zotero avancé), nous pourrons nous essayer à une utilisation plus avancée de Stylo.

Création d'un article & lien avec Zotero

#### **6.6 Rendu pour le 17 octobre**

Création d'un article à partir de l'article « La traduction de documents XML dans les mémoires de traduction » ([https://www.persee.fr/doc/equiv\\_0751-9532\\_2005\\_num\\_32\\_1\\_890](https://www.persee.fr/doc/equiv_0751-9532_2005_num_32_1_890)) de Pascaline Merten (Merten 2005) :

Méthode :

- parcourir le document et relevé les différentes tâches à réaliser (structuration, métadonnées) ;
- commencer par intégrer le texte en copiant-collant la « version texte » – à retrouver sur la page Persée –, puis en le structurant en vous référant à la version graphique : niveaux de titres, italiques, notes de bas de page, références bibliographiques, images ;
- ajouter au moins quelques références bibliographiques ;
- ajouter toutes les métadonnées dont vous disposez via le mode « Editor Mode ».

Une fois le travail terminé, partagez votre article avec moi ([mathilde.verstraete@umontreal.ca](mailto:mathilde.verstraete@umontreal.ca)) et [prévenez-moi par courriel](#).

# 7 Zotero

## 7.1 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.

## 7.2 Zotero : introduction



Tip

Lecture recommandée : (Daniel 2022).

### 7.2.1 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.

### 7.2.2 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.

### **7.2.3 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.

## **7.3 Utilisation de Zotero**

### **7.3.1 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).

### **7.3.2 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.

### **7.3.3 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).

### **7.3.4 Exemple à partir du catalogue de la Bibliothèque universitaire de l'UdeM**

À voir en cours.

### **7.3.5 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.

### **7.3.6 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.

### **7.3.7 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.

### 7.3.8 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.

### 7.3.9 Créer

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

## 7.4 Organiser

### 7.4.1 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 !

### 7.4.2 Tags et marqueurs

- Les tags et les marqueurs servent à ajouter des informations de description ;
- Vous pouvez créer votre propre système de marqueurs et de tags pour organiser plus finement vos références ;
- Certaines références importées comportent parfois des marqueurs, vous pouvez les inspirer ou les supprimer ;
- Une recherche dans Zotero interroge tous les champs, y compris les marqueurs et les tags !

### 7.4.3 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 !

## 7.5 Exercice

- 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 “HNU6051”.

## 7.6 Zotero avancé

## 7.7 Crédit de bibliographies

### 7.7.1 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. *Communication et organisation. Revue scientifique francophone en Communication organisationnelle*, 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 !

```
<div id="merzeau_editorialisation_2013">
    <dt>Merzeau (2013)</dt>
    <dd><span itemscope itemtype="https://schema.org/Article" data-type="article">
        <span itemprop="author" itemscope itemtype="https://schema.org/Person">
            <span itemprop="familyName">Merzeau</span>,
            <meta itemprop="givenName" content="Louise" /> L.</span>
            (<span itemprop="datePublished">2013</span>).
        <span itemprop="name">
            "Éditorialisation collaborative d'un événement"
        </span>.
        <meta itemprop="journal" <i>Communication et organisation. Revue scientifique francophone en Communication organisationnelle</i> <span>no 43, June 2013</span>.
    </span>
    </dd>
</div>
```

### 7.7.2 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 ;
- Crédit à 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](https://www.zotero.org/support/fr/creating_bibliographies)

### 7.7.3 Citer et créer une bibliographie directement dans un 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 modifiez vos références dans Zotero, vous pouvez mettre à jour votre document LibreOffice.

## 7.8 Crédit et gestion de groupes

### 7.8.1 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

## 7.9 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 bibliographique : chaque entrée correspond à une description d'un document avec des items « mot-clé = 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},
}
```

## 7.10 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).

### 7.10.1 Installez Better BibTeX

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

### 7.10.2 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

## 7.11 Questions ?

**Deuxième partie**

**session 2**

# 8 Le *single source publishing*



Tip

Lecture recommandée : (Fauchié and Audin 2023), ou à écouter (Ecrinum n.d.).

## 8.1 Kézako

Qu'est-ce que ça vous évoque, le *single source publishing* ?

## 8.2 Chaîne éditoriale

Ensemble des méthodes, des outils et des processus qui permettent de concevoir, fabriquer, produire et diffuser un livre ou plus largement une publication. Une chaîne de publication réunit des logiciels propres à la gestion du texte et des images, mais aussi les différents rôles d'un processus éditorial que sont l'auteur, l'éditeur, le correcteur, le graphiste, etc. (Fauchié 2024)

En général, idéal de la linéarité.

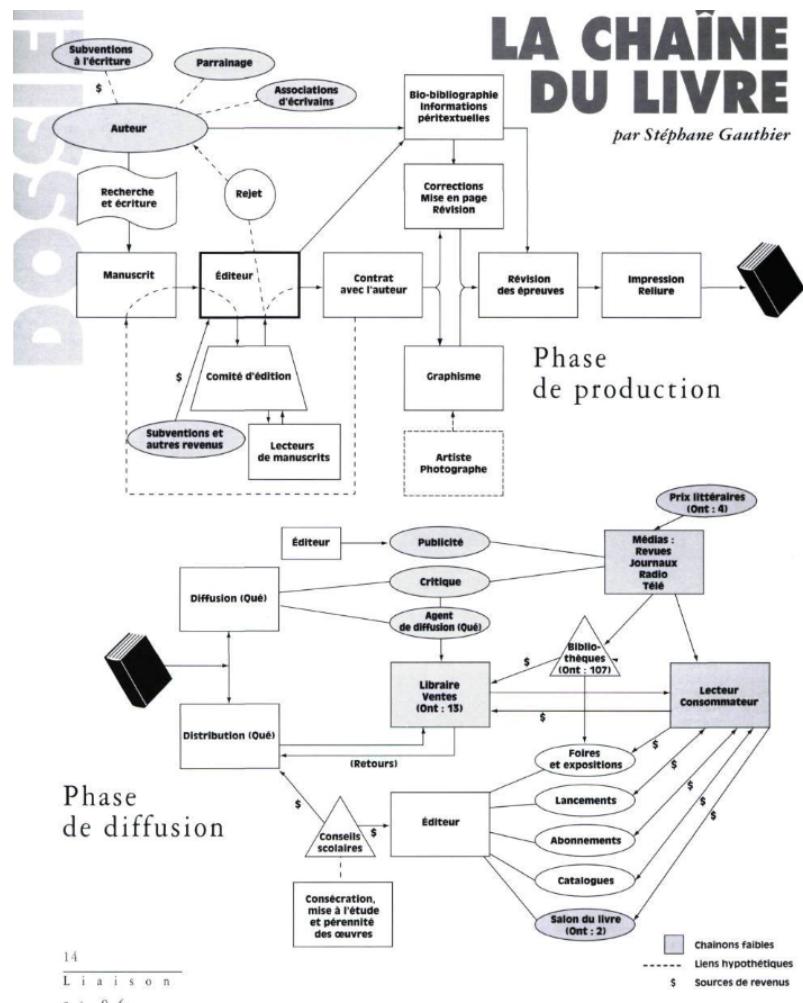


FIGURE 8.1 – Chaîne du livre (Stéphane Gauthier)

### 8.2.1 Les différentes étapes de la chaîne éditoriale type

- Soumission ou commande ;
- Lecture et retour du comité éditorial ;
- Nouvelle version ;
- Lecture et retour du comité de lecture / évaluation par les pairs ;
- Nouvelle version ;
- Édition ;
- Contrôle qualité ;
- Transmission à l'auteur ou Bon à tirer (BAT)

### **8.2.2 Et le numérique ?**

Réagencement de la vision linéaire de la chaîne éditoriale par des principes et des pratiques d'hybridation et de modularité

### **8.2.3 Rappel : plusieurs approches de la chaîne éditoriale**

classiques vs non-conventionnelles (modulaire)

Chaîne éditoriale classique

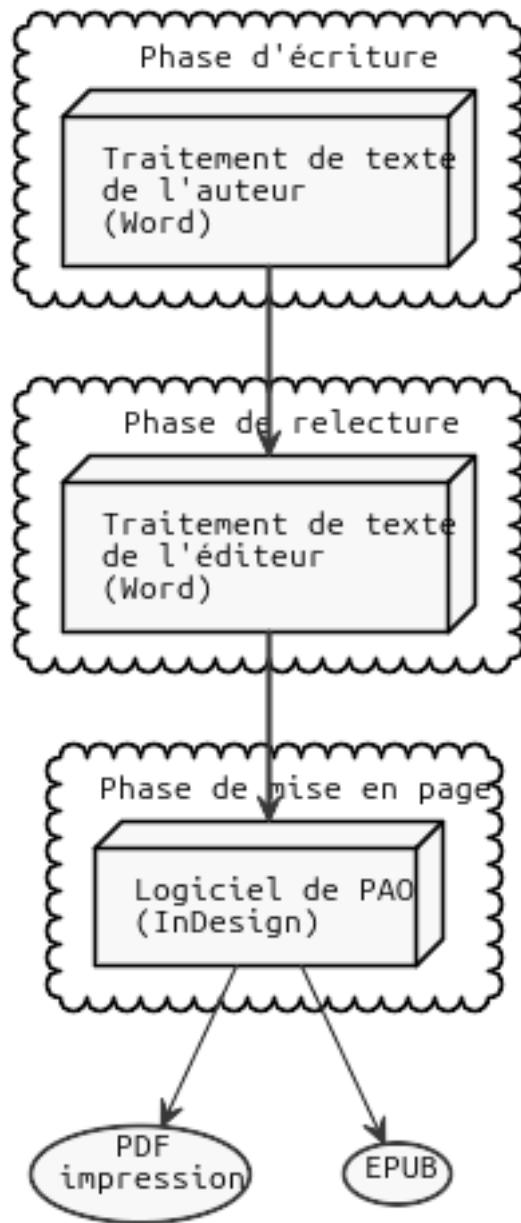


FIGURE 8.2 – Schéma Antoine Fauchié

Chaîne éditoriale modulaire

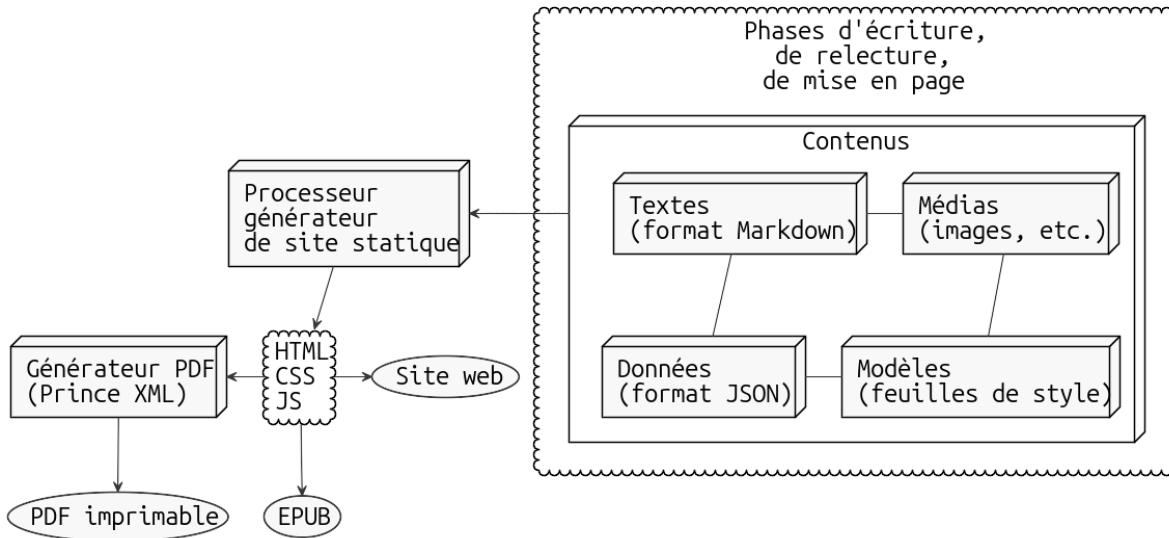


FIGURE 8.3 – Schéma Antoine Fauchié

### 8.3 Les chaînes d'édition modulaires

- Synergie entre les chaînons et interopérabilité technique (liée aux formats libres et ouverts) ;
- Exploration des caractéristiques des environnements numériques (pas de logiciels attribués) ;
- Agencement et adaptation selon les projets et les objectifs (construire une chaîne sur un objet).

### 8.4 Le *single source publishing* ou la publication à source unique

#### 8.4.1 Quelques termes

- Multisupport :

  - plusieurs supports compatibles avec la publication ;
  - papier ou numérique ;

- Multiformat :

  - plusieurs formats de publication ;
  - HTML, ePUB, PDF, etc. ;

- Multimodal :

  - plusieurs modes de lecture ;

— version PDF statique, version numérique enrichie/augmentée.

#### 8.4.2 Définition

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. (Fauchié)

#### 8.4.3 Ex. : un livre en SSP

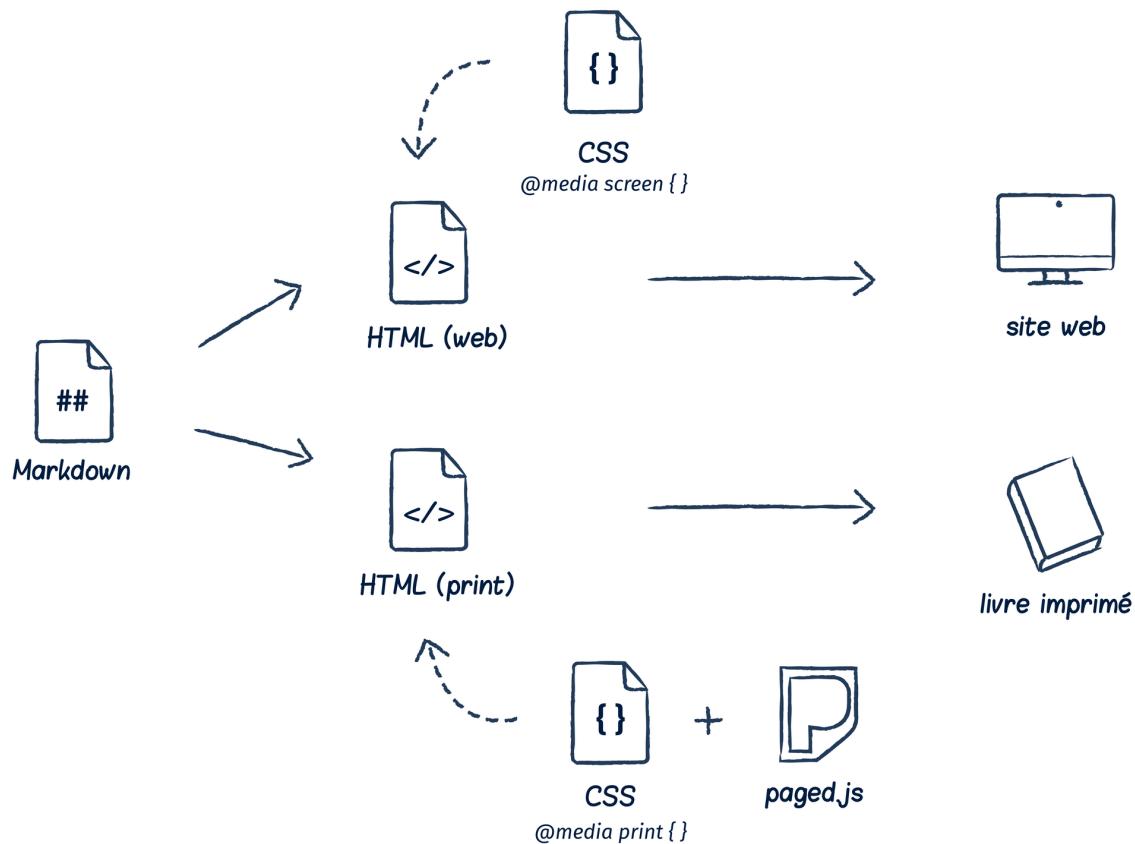


FIGURE 8.4 – Schéma Antoine Fauchié

#### 8.4.4 Ex. : un site en SSP

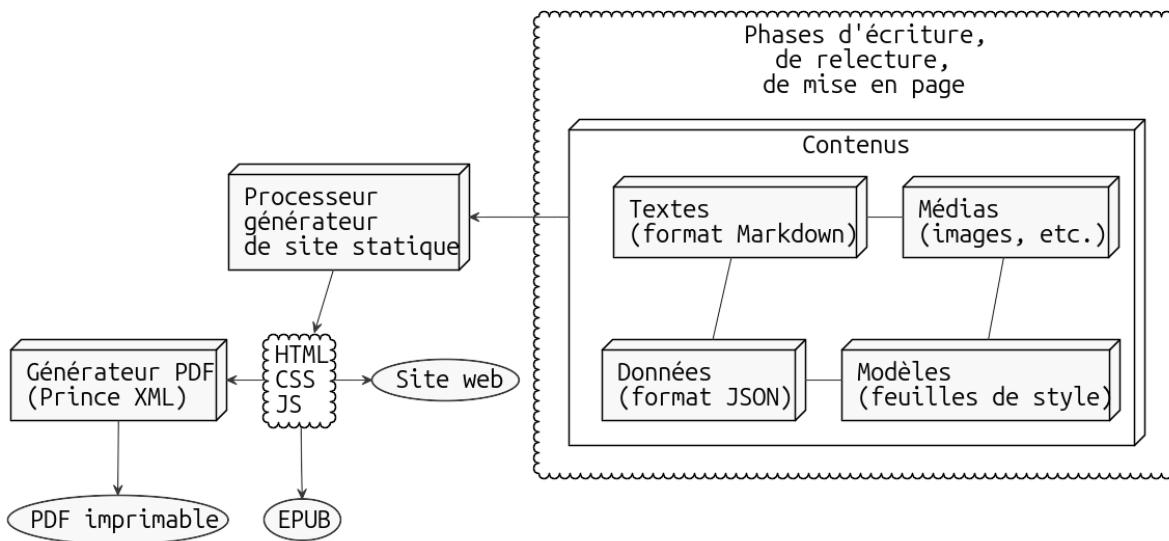


FIGURE 8.5 – Schéma Antoine Fauchié

#### 8.4.5 Intérêts et limites

- Clarification de la source de référence
- Mutualisation du travail
- Anticipation de l'archivage

mais...

- Un *ensemble* de fichiers de référence
- Temps et essais de la mise en place de la chaîne

### 8.5 Quelques exemples

- Stylo (cf (Vitali-Rosati et al. 2020)) : Markdown, YAML et BibTeX pour les données ; Pandoc et LaTeX pour la génération des formats ;
- [Quire](#) : Markdown, YAML et JSON pour les données ; Hugo et PrinceXML pour la génération des formats ;
- [Abriupt](#) : Markdown, HTML, YAML ; Make, Pandoc et LaTeX pour la génération des formats ;
- ...

# **9 Installation de *Codium***

## **9.1 C'est quoi *Codium* ?**

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

VSCode 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. Large-  
ment utilisé par la communauté des développeur.ses pour faire de la programmation.

## **9.2 Installation**

<https://vscode.com/#install>

Documentation officielle : <https://code.visualstudio.com/docs>

## **9.3 Extensions**

Suivre ce tutoriel (pour installer French Language Pack for VS Code, French - Code Spell Checker et LTeX, à configurer en français) :

<https://e-nsi.forge.aeif.fr/cours/1-general/VSCodium/#menus-en-francais>

# 10 Utiliser son terminal

## 10.1 Terminal, console, shell... ?

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*

-> Soit, l'*OS*, le « 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](#)).

### 10.1.1 Le *shell*

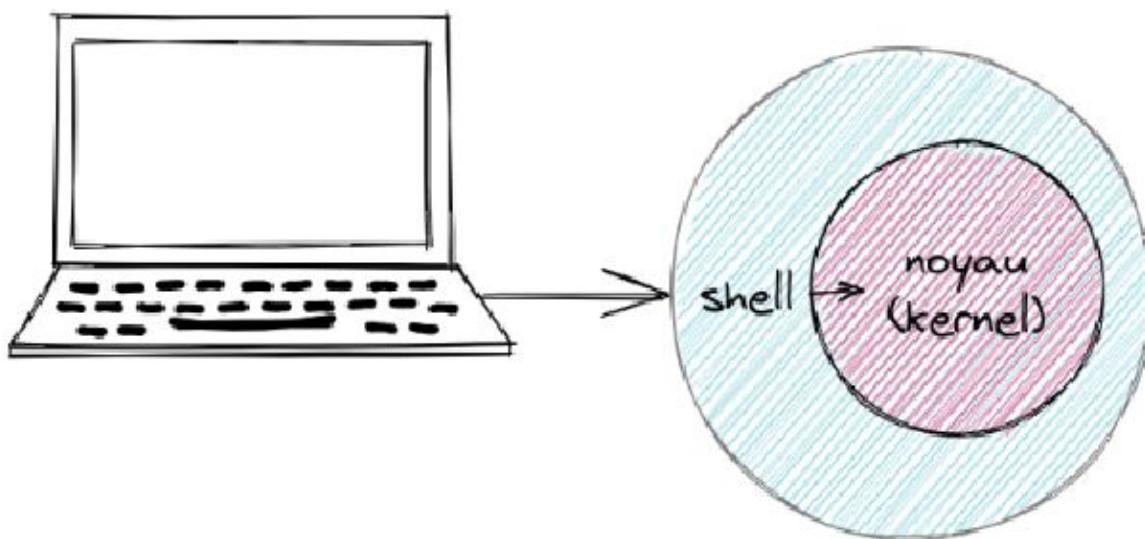


FIGURE 10.1 – Le shell, selon Alexis Janvier

### **Note**

Il existe plusieurs types d'interfaces humaines. Les plus courantes sont la **graphique**, la **formaté** et l'**écriture**.

#### **10.1.2 Terminal**

Le terminal, à l'origine, c'est un objet physique nous permettant d'interagir avec l'ordinateur :



FIGURE 10.2 – Terminal DEC VT100

#### **10.1.3 Émulateur de terminal**

- émuler une console système au sein d'un environnement graphique

### **Tip**

Lecture recommandée : (Drevetkenin 2021).

## **10.2 les CLI (*command-line interfaces*)**

Interface en ligne de commande permettant de dialoguer avec l'ordinateur en tapant des instructions textuelles (des **commandes**).

## **10.3 Intérêts**

- Rapidité et efficacité
- Automatisation (scripts)
- Contrôle avancé

## 10.4 Quelques commandes

Sous Linux ou Mac : chercher « Terminal » parmi les applications.

Sous Windows, chercher « Windows Power Shell » ; dans le menu déroulant en haut sélectionner « Ubuntu » ; écrire dans le terminal :

- pour savoir où l’on se trouve : `pwd`
- pour lister les fichiers : `ls`
- pour naviguer : `cd Documents/cours-hnu6051`
- pour comprendre à quoi sert une commande : `man nom-de-la-commande` (ex. : `man rm`)

### 10.4.1 Chemin absolu / chemin relatif

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
<code>/Users/Mathilde/Documents/Projet/index.html</code>	<code>/Documents/Projet/index.html</code>
Emplacement complet à partir de la racine du système de fichiers	Chemin vers un fichier à partir du répertoire actuel ( <code>pwd</code> )
Toujours valide, qu’importe le répertoire	Change selon où je me situe dans mes fichiers

### 10.4.2 Quelques bons 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.

### 10.4.3 Gérer des fichiers ou des dossiers

- `mkdir hnu6051`
- `cd hnu6051`
- `touch fichier-test.txt`

Pour trouver des fichiers qui contiennent un terme qui nous intéresse :

- `grep -ri mot-recherche`

Pour trouver tous les fichiers qui se terminent par l’extension `.txt` :

— “‘find . -type f -name “\*.txt”

#### 10.4.4 Déplacer un fichier



Tip

La commande `mv` permet de renommer ou de déplacer un fichier (dans le terminal, c'est la même chose !)

— `mv mon-fichier.txt fichier.txt`

— `mv /home/utilisateur/hnu6051/mon-fichier.txt /home/utilisateur/Documents/mon-fichier.txt`

### 10.5 Rendu pour le 18 octobre

Apprenez-moi quelque chose à propos du terminal ! Cela peut concerter une commande que vous trouvez utile, une astuce pour gagner du temps, ou une fonctionnalité du terminal que peu de gens connaissent. Nous commencerons le cours du 18 par vos trouvailles : une simple explication suffit, une démonstration est bienvenue.

Ex. : inscrivez dans votre terminal `curl http://wttr.in/Montréal\?lang\=fr`

# 11 LaTeX

## 11.1 LaTeX : introduction

- **Quoi** ? Un langage de composition de documents, principalement utilisé pour des textes scientifiques, techniques et mathématiques.
- **Quand** ? Créé au début des années 1980 par Leslie Lamport, basé sur TeX, inventé par Donald Knuth dès 1977.
- **Pourquoi** ? Fournir un moyen précis et reproductible de créer des documents complexes, (notamment des équations) :
  - Qualité typographique ++ (mise en page, figures, références croisées,...)
  - Gestion avancée des références bibliographiques
  - Flexibilité
  - Open-source et multiplateforme
  - Contenu vs Style (ou presque)

## 11.2 TeX, LaTeX, et al.

- **TeX** : Langage de base pour la composition de textes (inventé par Donald Knuth, cf. (Knuth 1984)) ;
- **LaTeX** : Interface simplifiée sur TeX, ajoutant des commandes prédéfinies pour faciliter la création de documents ;
- **LuaTeX/LuaLaTeX** : Extension moderne qui intègre le langage de programmation *Lua* pour des scripts plus flexibles ;
- **XeTeX/XeLaTeX** : Variante qui améliore la gestion des polices et des langues.

## 11.3 Structure générale

- Préambule : définit le type de document, les paquets utilisés, les configurations générales (marges, polices,...) ;
- Corpus du texte (`\begin{document}... \end{document}`) ;
- Compilation(s) du document -> PDF

## 11.4 Modules supplémentaires, *paquets*

Augmenter LaTeX via des *paquets* spécifiques :

- Modifier le fonctionnement de certaines parties de LaTeX ;
- Modifier l’apparence du documents ;
- Ajouter de nouvelles commandes ;
- ...

Ex. : `\usepackage{graphicx}`  
`\includegraphics[width=\textwidth]{image.png}`

## 11.5 Discussion : avantages, inconvénients

Qu’en pensez-vous, à première vue ?

## 11.6 Tuto

Commençons à réaliser ce tuto en classe : <https://www.learnlatex.org/fr/>

Vous devrez vous créer un compte sur [Overleaf](#)

La documentation officielle : <https://tug.org/begin.html>



Tip

Pour débuter, vous pouvez utiliser LaTeX sur Overleaf. Pour la suite, je vous conseille d’installer une distribution complète. Suivez les instructions d’installation ici : <https://tug.org/begin.html>.

**Troisième partie**

**session 3**

# 12 Des caractères à l'écran – encodage numérique

## 12.1 Encodage numérique

- Un fichier numérique = Une suite de bits (des 0 et des 1)
  - train/flux de bits : suite de bits (**binary digit**)
  - un octet :
    - une séquence de 8 bits
    - $2^8$  (256) combinaisons possibles de 0 et de 1
- Le format : spécifie l'interprétation de cette suite de 0 et de 1
- Taille d'un fichier numérique : nombre d'instances de 0 et de 1 nécessaires à sa représentation
- *Codage* : norme qui définit quels sont les caractères utilisables dans un fichier. Toutes les normes ne codent pas les mêmes caractères. (- A. Perret)

## 12.2 Système de numération en base 2

Un ordinateur ne connaît que deux choses : les 1 et les 0. Chaque lettre dans cette phrase, chaque couleur, chaque seconde d'une vidéo ou d'un morceau de musique, chaque page web, chaque programme n'est qu'une longue succession de 1 et de 0.

Mia Combeau

Décimal	Binaire
0	0
1	1
2	10 <-- 2 puissance 1
3	11
4	100 <-- 2 puissance 2
5	101
6	110
7	111
8	1000 <-- 2 puissance 3
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000 <-- 2 puissance 4
17	10001
18	10010
19	10011
20	10100
...	...
100	1100100
...	...
1000	1111101000
...	...
9000	10001100101000

## 12.3 Mémoire

- La taille d'un disque dur définit le nombre de bits disponibles pour encoder des informations simultanément. Par exemple, un disque dur de 50 Go de mémoire :
    - est égal à 50 000 000 000 octets (cinquante milliards)
    - est égal à 400 000 000 000 bits (quatre cents milliards)
  - La taille d'un fichier définit le nombre de bits nécessaire à son encodage. Par exemple, un fichier d'une taille de 30 Ko :
    - nécessite  $30 \times 1\,000 \times 8$  bits
    - contient donc un total de 240 000 instances de 1 ou 0

## 12.4 0 ou 1 ?

Tout document conservé sur un ordinateur est constitué d'une série de 0 et de 1.

Ce sont les normes d'encodage et les formats qui permettent d'interpréter le sens de ces 0 et 1.



La Joconde, ou Portrait de Mona Lisa, est un tableau de l'artiste Léonard de Vinci, réalisé entre 1503 et 1506 ou entre 1513 et 1516,2, et peut-être jusqu'à 1519.3 L'artiste était mort cette année-là, le 2 mai 1519, qui représente probablement la date de la mort, probablement celle de la Florentine Lisa Gherardini, épouse de Francesco del Giocondo. Acquise par François Ier, cette peinture à l'huile sur panneau de bois de peuplier de 77 x 53 cm est exposée au musée du Louvre à Paris. La Joconde est l'un des rares tableaux attribués de façon certaine à Léonard de Vinci.

La Joconde est devenue un tableau éminemment célèbre car, depuis sa réalisation, nombre d'artistes l'ont pris comme référence. À l'époque romantique, les artistes ont été fascinés par ce tableau et ont contribué à développer le mythe qui l'entoure, en faisant de ce tableau l'une des œuvres d'art les plus célèbres du monde, si ce n'est la plus célèbre : elle est en tout cas considérée comme l'une des représentations d'un visage féminin les plus célèbres au monde. Au XXIe siècle, elle est devenue l'objet d'art le plus visité au monde, devant le diamant Hope5, avec 20 000 visiteurs qui viennent l'admirer et la photographier quotidiennement6.

## 12.5 Normes d'encodage

- Normes d'encodages = clefs d'encodage/décodage pour passer d'une forme intelligible pour les humains à une forme lisible par la machine ;
  - En fonction du type de fichier encodé (texte, image, vidéo, etc.), il existe différents paysages de normes et de modes de représentation

## 12.6 Encodage des données textuelles

- Un document numérique textuel contient du texte (caractères alphanumériques et de symboles) ;
- La norme d'encodage d'un texte == une table de correspondance entre un groupe de bits et un caractère alphanumérique ou un symbole ;
- La norme d'encodage définit donc la liste des caractères disponibles (**character set**) ;
- En fonction de la norme, la représentation d'un caractère peut s'étendre sur un ou plusieurs octets
- Les normes les plus connues pour des textes en français sont :
  - ASCII (1963)
  - ISO 8859-1 (1986)
  - UTF-8 (1996)

### 12.6.1 ASCII

- [American Standard Code for Information Interchange](#) ;
- Premier jeu de caractères standardisé (1963) ;
- Norme américano-centrée ;
- 128 caractères, desquels sont exclus :
  - les caractères accentués,
  - tous les alphabets non latins (arabe, cyrillique, hébreu, idéogrammes, etc.) ;
- 1 caractère = 7 bits

```
! "#$%& ' ()*+, -./
0123456789: ;<=>?
@ABCDEFGHIJKLMNO
PQRSTUVWXYZ[\]^_
`abcdefghijklmno
pqrstuvwxyz{|}~
```

**i** Note

Ci-dessus, seulement 95 caractères. Les autres ne sont pas imprimables (dits, caractères *de contrôle*) ; il correspondent plutôt à des actions ou des commandes.

Ex. :

- 9 (binaire : 000 1001) = HT (*Horizontal Tab, tab*)
- 10 (binaire : 000 1010) = LF (*Line Feed, saut de ligne*)
- 12 (binaire : 000 1100) = FF (*Form Feed, saut de page*)
- 127 (binaire : 111 1111) = à votre avis ?

## 12.6.2 ISO 8859-1

- 1986
  - 1 caractères = 8 bits
  - Rétrocompatibilité ASCII ;
  - 256 caractères où chaque caractère est représenté par 8 bits ;
  - Souvent appelée Latin1 mais aussi Western Europe.
  - Manquent : œ, € (et les alphabts non-latins)

!"#\$%&'()-\*,-./  
0123456789:;=>?  
@ABCDEFGHIJKLMNO  
PQRSTUVWXYZ[\]^\_  
`abcdefghijklmnno  
pqrstuvwxyz{|}~

- Exemple :  $e = 11101001$

## i Note

- ISO 8859-2 (Europe centrale et de l'Est),
  - ISO 8859-5 (alphabet cyrillique),
  - ISO 8859-6 (caractères courants de l'alphabet arabe), etc.

où seuls changent les codes 128 à 255.

### 12.6.3 Unicode

- Table de caractères ;
  - Standard universel de représentation de caractères ;
  - Contrairement à ASCII, Unicode ne définit pas l'implémentation du *mapping* ;
  - Indique seulement quel caractère se réfère à quel point de code (où chaque point de code est un nombre hexadécimal (0-9 + A-F) représentant un caractère) ;
  - Le standard Unicode lui-même ne spécifie pas comment ces points de code doivent être stockés en mémoire ou transmis dans des fichiers. C'est là qu'interviennent les encodages Unicode comme UTF-8 ;
  - 1 114 112 positions possibles ;
  - Divisés en *scripts* : collections cohérentes de caractères en usage dans un domaine particulier (symboles monétaires, opérateurs mathématiques, braille, etc.).

#### 12.6.4 UTF-8

- Méthode d’encodage
- 1996
- [Universal Character Set Transformation Format](#) + base d’encodage (8, 16, 32, ...), soit UTF-8, UTF-16, UTF-32 ;
- En fonction de la base d’encodage, un caractère peut être représenté par 1, 2, 3 ou 4 octets ;
- UTF est rétrocompatible avec ASCII, mais pas avec ISO 8859-1. Il s’inscrit dans le cadre établi par Unicode qui vise à faciliter l’échange de textes numériques en assurant leur compatibilité ;
- Unicode définit plus de 137 000 caractères, issus de multiples alphabets, et prévoit aussi des symboles comme les emojis ;
- Exemples
  - A = U+0041 = 01000001 ;
  - = U+6C49 = 11100110 10110001 10001001 ;
  - = U+1F913 = 11110000 10011111 10100100 10010011 ;

### 12.7 Mauvais encodage/décodage

Si on n’utilise pas la bonne clef d’encodage ou de décodage d’un texte, on risque d’obtenir un texte partiellement ou entièrement illisible.

UTF-8 ouvert avec 8859-1 (Latin 1)

La Joconde, ou Portrait de Mona Lisa, est un tableau de l’artiste Léonard de Vinci, réalisée entre 1503 et 1506 ou entre 1513 et 1516, et peut-être jusqu’à 1519 (l’artiste étant mort cette année-là, le 2 mai)3, qui représente un portrait mi-corps, probablement celui de la Florentine Lisa Gherardini, épouse de Francesco del Giocondo. Acquise par François Ier, cette peinture à l’huile sur panneau de bois de peuplier de 77 × 53 cm est exposée au musée du Louvre à Paris. La Joconde est l’un des rares tableaux attribués de façon certaine à Léonard de Vinci.

UTF-8 ouvert avec ISO 8859-5 (Cyrillic)

La Joconde, ou Portrait de Mona Lisa, est un tableau de l’artiste Léonard de Vinci, réalisé entre 1503 et 1506 ou entre 1513 et 1516, et peut-être jusqu’à 1519 (l’artiste étant mort cette année-là, le 2 mai)3, qui représente un portrait mi-corps, probablement celui de la Florentine Lisa Gherardini, épouse de Francesco del Giocondo. Acquise par François Ier, cette peinture à l’huile sur panneau de bois de peuplier de 77 × 53 cm est exposée au musée du Louvre à Paris. La Joconde est l’un des rares tableaux attribués de façon certaine à Léonard de Vinci.

## 12.8 Une norme ?

UTF s'impose désormais comme la norme

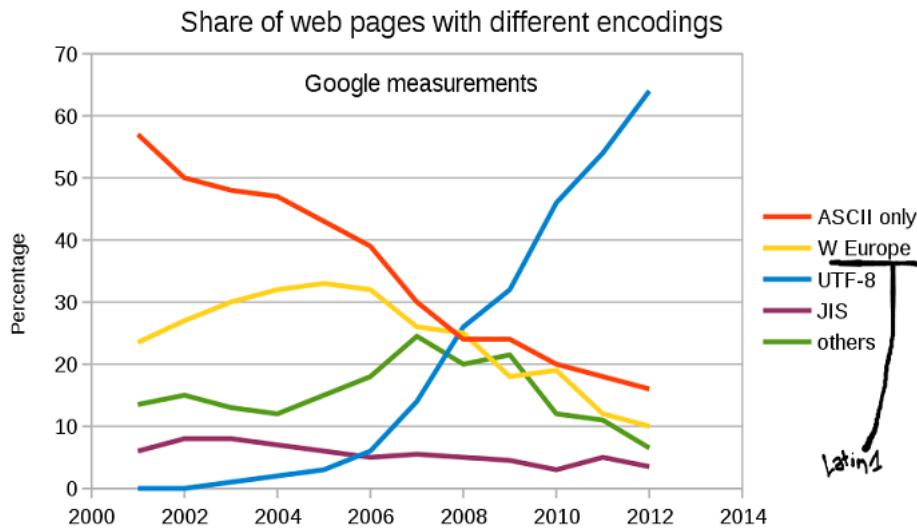


FIGURE 12.1 – diagramme montrant qu'UTF s'impose sur les autres normes depuis 2006

D'après W3Techs, 97,8% des pages web qui déclarent un encodage suivent UTF-8 aujourd'hui.

(source du graphique : Wikimedia)

# 13 Convertisseurs : le cas de *pandoc*

Fabriquer sa propre chaîne de publication depuis une source unique

## 13.1 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>

## 13.2 Fonctionnement (en ligne de commande)

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`)

### 13.3 Installation (en classe)

Ce lien <https://pandoc.org/installing.html> vous amène directement aux dernières *releases* sur GitHub. Téléchargez le système qui convient le mieux à votre système d’exploitation, soit :

- Pour Windows : `pandoc-3.4-windows-x86_64.msi` ;
- Pour Mac :
  - Mac avec puces Intel (x86\_64) : `pandoc-3.4-x86_64-macOS.pkg` ;
  - Macs avec puces Apple Silicon (ARM, M1/M2) : `pandoc-3.4-arm64-macOS.pkg` ;
  - Vous hésitez ? `Cmd + Espace`, *Informations Système*. Dans la colonne de gauche, sélectionner Matériel. La ligne “Nom du processeur” ou “Type de processeur” apparaîtra dans la section principale.

### 13.4 Quelques options de la commande Pandoc

- `--standalone` (produit un document complet incluant les métadonnées) ;
- `--bibliography=references.bib` (spécifie un fichier de références bibliographiques) ;
- `--template=mytemplate.html` (spécifie un modèle) ;
- `--toc` (table of contents) ;
- `--help` (résume les options existantes)

### 13.5 Exemples supplémentaires

1. `.docx -> .md`

```
pandoc -f docx -t markdown --extract-media=./ fichier.docx -o fichier.md,  
où :
```

- `--extract-media=./` : extraction et sauvegarde des fichiers multimédias dans le répertoire courant ;

2. `.md -> .html`

```
pandoc --standalone --ascii --bibliography=references.bib -f markdown -t html  
fichier.md -o fichier.html, où :
```

- `--standalone` : production d’un document `.html` autonome (soit, ajout des balises `<html>` ; `<head>` ; `<body>`) ;
- `--ascii` : utilisation des caractères ASCII uniquement ;
- `--bibliography=references.bib` : intégration d’une bibliographie à partir du fichier `references.bib`

## 13.6 Métadonnées (exercice)

Ajoutez les métadonnées en en-tête d'un fichier .md :

```
---  
title: "mon titre"  
subtitle: "mon sous-titre"  
date:  
author:  
- Aristotle  
- Plato  
abstract:  
keywords:  
bibliography:  
lang:  
---
```

### 💡 Tip

Ces métadonnées peuvent être placées soit en en-tête d'un fichier .md, soit dans un fichier séparé, au format YAML (extension : .yaml). Dans ce cas, la commande Pandoc précédente devra préciser où sont les métadonnées :

```
pandoc --standalone -f markdown -t html monfichier.md metadonnees.yaml -o export.html
```

## 13.7 Export d'une bibliographie

— En ligne de commande :

```
pandoc --standalone --bibliography=monfichier.bib --filter pandoc-citeproc -f markdown -t html monfichier.md -o export.html
```

— Dans le fichier principal (dans l'en-tête yaml) ; lancer la même commande que ci-dessus, sans --bibliography=monfichier.bib :

```
---  
title: "mon titre"  
subtitle: "mon sous-titre"  
...  
bibliography: monfichier.bib  
---
```

 Tip

1. Ajouter `link-citations: true` dans le yaml permet de transformer les appels de références en hyperliens.
2. Pour ajouter tous les éléments du `.bib`, même ceux non cités, ajoutez en yaml : `nocite: '@*'`

## 13.8 Les *templates*

- Template par défaut pour chacun des formats de sorties ;
- Principe : Pandoc remplace les variables du template par les valeurs déclarées dans les métadonnées `.yaml` (fichier à part ou en-tête)
- Par ex. :

yaml	Template	html (par ex.)
<code>author: Platon</code>	<code>&lt;meta name="author" content="\$author\$"&gt;</code>	<code>&lt;meta name="author" content="Platon"</code>

## 13.9 Personnaliser son *template* (exercice)

1. Récupérer le template par défaut d'un format :

```
pandoc -D html > montemplate.html
```

 Tip

Cette commande fonctionne pour tous les formats de sortie (hormis `.docx/.odt`, cf. la documentation).

2. Ouvrez `montemplate.html` dans VSCode pour voir comment il est structuré.
3. Dans les données yaml d'un document `.md`, ajoutez la ligne `image: auteur.jpg`
4. Dans le template, nous pouvons ajouter une balise :

```
$if(image)$  $endif$
```

5. Pour que Pandoc prenne en compte le nouveau template, ajoutez l'option `--template=FILE`, soit :

```
pandoc --standalone --filter pandoc-citeproc --template=montemplate.html -f markdown -t html monfichier.md metadata.yaml -o export_2.html
```

### **13.10 Questions ?**

# 14 Quarto

## 14.1 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,...

<https://quarto.org/about.html>

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.  
[\(Quarto\)](#)

## 14.2 Fonctionnement

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

Intégration de code (Python, R, JavaScript)

## 14.3 Historique

RStudio (devenu Posit) : environnement de développement gratuit, libre et multiplateforme pour R, un langage de programmation utilisé pour le traitement de données et l'analyse statistique

Format : RMarkdown, .rmd (qui ressemble au .qmd de QuartoMarkdown)

## 14.4 Au travail

1. Installez Quarto : <https://quarto.org/docs/get-started/> ;
2. Faites le tuto *Hello, Quarto* : <https://quarto.org/docs/get-started/hello/text-editor.html> ;
3. Faites le tuto *Authoring* : <https://quarto.org/docs/get-started/hello/text-editor.html> ;
4. Si le temps le permet, commencez votre rendu, la documentation est ici : <https://quarto.org/docs/books/> + versionnez votre projet sur GitHub ;
5. Pour la publication en ligne, consultez cette page : <https://quarto.org/docs/publishing/github-pages.html> (je conseille de suivre *Publish Command* : <https://quarto.org/docs/publishing/github-pages.html#publish-command>)

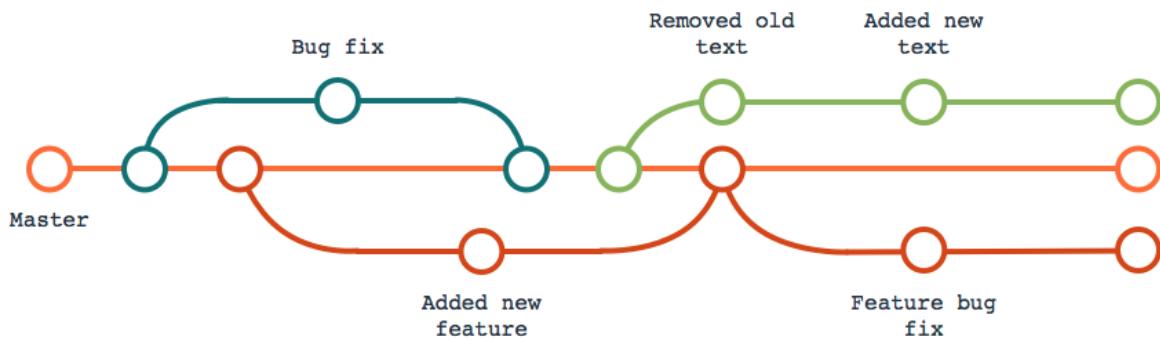
# 15 Git et GitHub

## 15.1 Versionnement

Git est un outil de versionnement et de synchronisation de fichiers.

Intérêts :

- gestion de nombreuses versions de plusieurs fichiers ;
- éviter les `memoire-2024-reluJeanJacques-version-finale-ok-ultime-v4-2024-10-18.docx` ;
- collaboration
- Établir une version du texte + travailler sur ses états



## 15.2 Git vs Github

Git : Logiciel de gestion de versions décentralisé libre et gratuit (installé sur votre ordinateur)

GitHub : Service web d'hébergement et de gestion de développement de logiciels (service en ligne)

### 15.2.1 Git

- Système de gestion de versions décentralisées créé pour gérer des fichiers textes de projets informatiques ;
- (Très) complexe, mais utilisation basique possible ;

- Conservation de toutes les versions des fichiers d’un projet à chaque enregistrement ;
- Git est utilisé sur des plateformes comme GitHub ou GitLab qui en facilite (grandement) l’usage.

### 15.2.2 GitHub

Service web d’hébergement qui utilise le logiciel de versions Git

- Propriété : Microsoft ;
- Alternatives libres : GitLab, Framagit

### 15.2.3 Fonctionnement

- Branches : comparer, fusionner, suivre ;
- Local / remote :
  - pull : pour récupérer les modifications de la version en ligne ;
  - push : pour publier les modifications vers la version en ligne ;
- Description des modifications

## 15.3 Quelques commandes de base

- `git clone [répertoire]` : clone un “répertoire distant” sur la machine locale dans un “répertoire local”, par exemple depuis le serveur framagit.org. Exemple : `git clone https://framagit.org/marviro/tutorielmdpandoc.git` ;
  - `git add fichier.md` : ajoute **un** fichier à l’index pour préparer un “commit” ciblé ;
  - `git add .` : ajoute **tous** les fichiers à l’index pour préparer un “commit” ;
- `git commit -m "message pour documenter la modification"` : valide un commit avec un message d’information, et donc déclare et inscrit les modifications au registre Git (enregistrement de l’état du projet).
  - `commit = label` de la version complété par l’utilisateur pour décrire la modification effectuée
- `git pull` : synchronise le répertoire local depuis le répertoire distant, ce qui permet de récupérer des modifications faites par d’autres contributeurs (on *tire* les modifications des autres depuis le serveur).
- `git push` : synchronise le répertoire distant à partir du répertoire local, ce qui permet de partager son travail avec d’autres contributeurs (on *pousse* ses modifications vers le serveur).
- `git status` : présente un état du répertoire : quels fichiers ont été ajoutés, modifiés, supprimés, non ajoutés au registre, etc.

### 15.3.1 Quelques commandes un peu plus complexes

- `git diff` : présente les changements de chaque fichier modifié.
- `git log` : présente une liste de commit. C'est l'état du registre, une forme d'historique des modifications.
- `git checkout` : changement ou création de branches, ce qui permet d'avoir une version dupliquée du projet ;
- `git rebase` ou `git merge` : fusion de plusieurs versions d'un même fichier (par exemple en réunissant plusieurs branches parallèles).

## 15.4 Git chatons

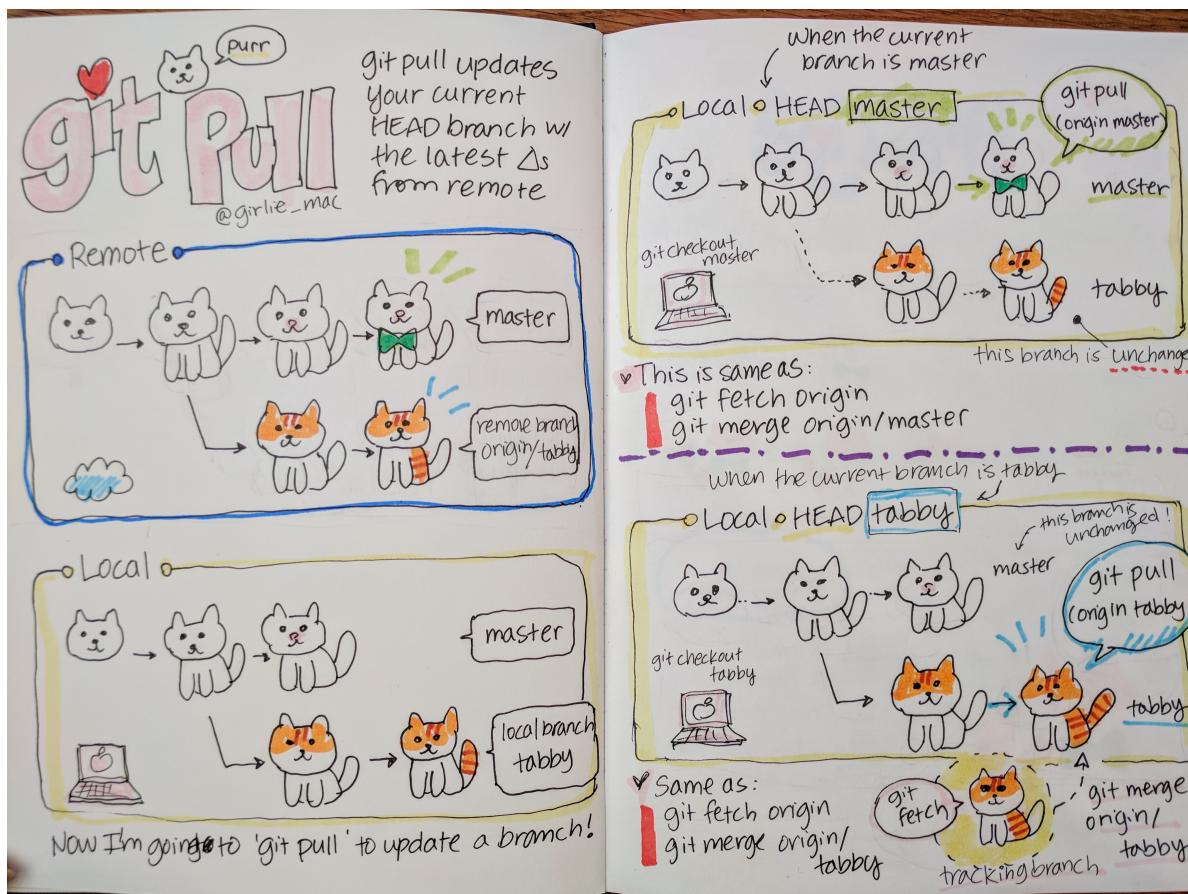


FIGURE 15.1 – Tomomi Imura (<https://girliemac.com/>)

## 15.5 Exercice - créer son premier répertoire

Vous allez créer un premier répertoire. Allons-y pas à pas :

1. ouvrez un terminal.
  - Par défaut, le terminal s’ouvre dans votre dossier personnel.
  - Lancez la commande `ls` pour vous en assurer.
2. créer un dossier GitHub : `mkdir github` et naviguez dedans `cd github`. Le terminal est maintenant positionné dans le répertoire `github`.
3. dans votre navigateur, allez sur votre compte GitHub puis cliquer sur [New Project]. Renseignez le *Project name* avec votre prénom (sans accents), puis cochez la case “Initialize repository with a README”, et enfin le bouton [Create project].
4. vous venez de créer un projet git. Vous allez maintenant le cloner, c'est-à-dire récupérer le répertoire sur votre machine en local. Pour cela, commencez par copier-coller l'url de clonage en cliquant sur le bouton bleu [Clone] en haut à droite, et en copiant l'url HTTPS.
5. dans le terminal, écrivez la commande `git clone` puis coller l'url copiée, et lancez la commande complète, par exemple : `git clone https://github.com/monprenom/monprojet.git`
6. ...
6. Git télécharge le répertoire et crée un nouveau répertoire [monprenom] dans le répertoire GitHub. Naviguez dans ce nouveau répertoire `cd monprenom`.
7. `ls` ou `ls -a` ou encore `ll` permet de lister tous les fichiers du répertoire. Pour le moment, il n'y a pas grand chose.
8. nous allons récupérer les sources de votre article édité sur Stylo en lançant un export ZIP. Une fois le dossier compressé téléchargé, décompressez le et copier les trois fichiers dans votre répertoire [monprenom]. Renommez les fichiers selon une bonne nomenclature.
9. nous pouvons maintenant ajouter ces fichiers à l'index Git : `git add .`, puis générer un premier commit pour ajouter ces fichiers au registre : `git commit -m "initialisation de l'article"`. On peut vérifier le registre avec `git log`.
10. essayons maintenant de synchroniser les modifications en local avec le serveur : `git push`. Sur le site de GitHub, votre répertoire est maintenant mis-à-jour.

Vous êtes maintenant en mesure de faire des modifications et de les synchroniser avec GitHub. Pendant le travail d'édition, vous pouvez régulièrement valider les modifications et synchroniser avec le serveur. Il suffit pour cela de répéter les opérations 9 et 10 : - `git add .` - `git commit -m "message"` - `git push`

## 15.6 Installation de Git

<https://git-scm.com/downloads>

# 16 YAML

## 16.1 Définition

- Yet Another Markup Language (yaml 1.0)
- YAML Ain't Markup Language (yaml 1.1)
- Format de représentation de données
- Langage de programmation, facile à lire et à comprendre
- Utilisable avec d'autres langages de programmation
- Pratique pour définir des métadonnées et configurer des fichiers

## 16.2 Syntaxe de base

- Liste de paires “clé-valeur”

```
clé: valeur
clé2: valeur2
```

- Importance de l'indentation (= hiérarchie) ; préférez les espaces aux tab
- Listes : un tiret pour chaque élément
- Commentaires : les lignes commençant par # ne sont pas interprétées

```
Author:
  - Platon
  - Socrate
# Mince, demander à Socrate s'il veut publier aussi
Place: "Athens"
```

## 16.3 Quarto et yaml

Quarto fonctionne avec des fichiers .mdoù yaml est utilisé principalement pour définir des métadonnées en en-tête (= *front matter*) Il est placé entre ---

```
---  
title: "Mon Document"  
author: "Mathilde"  
date: "2024-10-17"  
format: html #format de sortie souhaité  
---
```

## 16.4 Utilisation avancée

```
---  
title: "Titre"  
author: "Mathilde"  
date: "2024-10-17"  
bibliography: references.bib  
format:  
  pdf:  
    toc: true # Ajoute une table des matières  
    number-sections: true # Numérote les sections  
  html  
---
```

## 16.5 Exercice

Ceci me provoque une erreur, pourquoi ?

```
---  
title: "Le "Banquet" de Platon, analyse"  
author: "Mathilde"  
date: "2024-10-17"  
bibliography: references.bib  
format: pdf  
---
```

# Références

- Boulétreau, Viviane, and Benoît Habert. 2014. “Les formats.” In *Pratiques de l'édition numérique*. Les Ateliers de [sens public].
- Coombs, James H., Allen H. Renear, and Steven J. DeRose. 1987. “Markup Systems and the Future of Scholarly Text Processing.” *Commun. ACM* 30 (11) : 933–47. <https://doi.org/10.1145/32206.32209>.
- Daniel, Johanna. 2022. “Ode à Zotero, une décennie d’usage de Zotero.” {Billet}. *Isidore & Ganesh*. <https://doi.org/10.58079/pw5v>.
- Drevekenin, Aram. 2021. “Anatomy of a Terminal Emulator.” *Poor.dev*.
- Eberle-Sinatra, Michael, and Marcello Vitali Rosati. 2014. *Pratiques de l'édition numérique*. Parcours numériques 1. Montréal : Les Presses de l’Université de Montréal.
- Ecrinum. n.d. “Publication à source unique - Antoine Fauchié.” Accessed October 9, 2024.
- Fauchié, Antoine. 2017/2018. “Markdown Comme Condition d'une Norme de l'écriture Numérique.” *Réel-Virtuel*, no. 6 (2017/2018).
- . 2024. “Fabriquer des éditions, éditer des fabriques : reconfiguration des processus techniques éditoriaux et nouveaux modèles épistémologiques.” PhD thesis, Montréal : Université de Montréal.
- Fauchié, Antoine, and Yann Audin. 2023. “The Importance of Single Source Publishing in Scientific Publishing.” *Digital Studies / Le Champ Numérique*, no. Special DSCN Collection #9 (October). <https://doi.org/10.16995/dscn.9655>.
- Knuth, Donald E. 1984. *The TeXbook*. 1. édition. Reading, Mass. : Addison-Wesley Professional.
- Merten, Pascaline. 2005. “La traduction de documents XML dans les mémoires de traduction.” *Equivalences* 32 (1) : 63–78. <https://doi.org/10.3406/equiv.2005.890>.
- Merzeau, Louise. 2013. “Éditorialisation collaborative d'un événement : L'exemple des Entretiens du nouveau monde industriel 2012.” *Communication & Organisation* 43 (1) : 105–22. <https://doi.org/10.4000/communicationorganisation.4158>.
- Rouquette, Maïeul. 2012. *(Xe)LaTeX appliqué aux sciences humaines*. Atramenta.
- Simpkin, Sarah. 2015. “Débuter avec Markdown.” *Programming Historian*, November.
- Vitali-Rosati, Marcello, Nicolas Sauret, Antoine Fauchié, and Margot Mellet. 2020. “Écrire Les SHS En Environnement Numérique. L’éditeur de Texte Stylo.” *Revue Intelligibilité Du Numérique*, no. 1. [https://doi.org/10.34745/NUMEREV\\_1697](https://doi.org/10.34745/NUMEREV_1697).