

# Markdown pour les SHS

**Tuto@Mate – 06 février 2023**

**Daphné MATHELIER** | [daphne.mathelier@cnrs.fr](mailto:daphne.mathelier@cnrs.fr)

Service Humanités numériques (MSH Mondes)

Correspondante MSH de l'IR\* Huma-Num

*Programme* —————

1. Découvrir Markdown
2. Écrire en Markdown
3. Exemples d'utilisation

*Outils évoqués* —————

Gitlab , HedgeDoc , Hugo , Jupyter Notebook , Marp , reveal.js , R Markdown  
RStudio , Stylo , ShareDocs , Obsidian , VSCodium , Zettlr , Zotero



# 1 – Découvrir Markdown

## Qu'est-ce que Markdown ?



Markdown est un **langage de balisage léger** facile à lire et à écrire créé en 2004 par **John Gruber** (avec l'aide d'**Aaron Swartz** pour la syntaxe)

Philosophie de Markdown (**selon son créateur**) :

*Un document formaté en Markdown doit pouvoir être publié tel quel, en texte brut, sans donner l'impression d'avoir été structuré par des balises ou des instructions de formatage. [...]*

*La syntaxe Markdown est conçue dans un seul but : être utilisée comme format d'écriture pour le web.*

```
# Un titre (titre de niveau 1)
```

```
## Un sous-titre (titre de niveau 2)
```

```
### Une section (titre de niveau 3)
```

On peut formater du texte **en gras**, en *italique* ou ***les deux*** !

Une liste non numérotée :

- élément 1
- élément 2
- élément 3

Une liste numérotée :

1. élément 1
2. élément 2
3. élément 3

Pour changer de paragraphe il faut laisser une une ligne blanche.

On peut mettre des commentaires avec entre les balises `<!--` et `-->` et des hyperliens : [passage cliquable](#)(URL)

On peut mettre `des extraits de code` sur un ligne ou dans un bloc de code `<!-- chunk -->` :

```
```python
print("toto")
```
```

## Un titre (titre de niveau 1)

### Un sous-titre (titre de niveau 2)

#### Une section (titre de niveau 3)

On peut formater du texte **en gras**, en *italique* ou ***les deux*** !

Une liste non numérotée :

- élément 1
- élément 2
- élément 3

Une liste numérotée :

1. élément 1
2. élément 2
3. élément 3

Pour changer de paragraphe il faut laisser une une ligne blanche.

On peut mettre des commentaires avec entre les balises `<!--` et `-->` et des hyperliens : [passage cliquable](#)

On peut mettre `des extraits de code` sur un ligne ou dans un bloc de code :

```
print("toto")
```



## Avantages du Markdown

- **Simplicité** : facile à apprendre, lire et écrire ; à copier, partager et conserver
- **Portabilité** : multiplateforme / support / programme ; document pivot convertissable
- **Modernité** : écriture numérique et académique ; minimalisme et sobriété numérique

## Le format Markdown

- **Fichier texte brut**, léger, fiable, interopérable, pérenne
- **Lecture/écriture** dans tous les éditeurs de texte/code
- **Langage de balisage** qui permet la structuration sémantique d'un document
- **Syntaxe minimaliste** proche du langage naturel et qui va à l'essentiel
- **Communauté active** en augmentation constante depuis 2004
- **Adoption** par de grands acteurs du numérique

## Markdown comme alternative à Word !

→ Une solution pour sortir de l'usage exclusif, immodéré, voire inadapté de Word et Cie

Markdown est une alternative à connaître et qui permet :



- De se concentrer sur l'écriture d'un texte
- De se libérer des distractions du traitement de texte
- De visualiser sa structuration sémantique et logique
- De choisir la solution la plus adaptée à son besoin
- De ne pas subir les injonctions de format (grâce aux exports)

*Utiliser Markdown [...] permet de revenir à des méthodes qui vont à l'essentiel en faisant les choses dans l'ordre – écrire, illustrer, éditer puis publier – de manière durable et sans perte de temps et d'énergie inutiles. Pochet 2022*

## En bref

| Word                              | Markdown                          |
|-----------------------------------|-----------------------------------|
| Structuration esthétique (WYSWYG) | Structuration sémantique (WYSWYM) |
| Normes de l'imprimé (page)        | Normes du numérique (flux)        |
| Format propriétaire               | Format ouvert                     |
| Abonnement                        | Gratuit                           |
| Fichier intermédiaire             | Fichier pivot                     |
| Code bruyant                      | Code minimaliste                  |
| Conteneur de fichiers             | Fichier unique et léger           |
| Usages définis                    | Personnalisations possibles       |

## 2 - Écrire en Markdown

| Markdown                      | HTML                                    | Rendu   |
|-------------------------------|---|---|
| # Titre 1                     | <h1>Titre1</h1>                         | Titre 1   |
| *italique* OU <u>italique</u> | <em>italique</em>                       | <i>italique</i>   |
| **gras**                      | <strong>gras</strong>                   | <b>gras</b>   |
| ***gras italique***           | <strong><em>gras italique</em></strong> | <b><i>gras italique</i></b>   |
| XIX <sup>e</sup>              | XIX<sup>e</sup>                         | XIX <sup>e</sup>  |
| H <sub>2</sub> O              | H<sub>2</sub>O                          | H <sub>2</sub> O  |
| [hyperlien](url)              | <a href="url">hyperlien</a>             | <a href="#">hyperlien</a>   |
| ![alt text](image.jpg)        |     |  |
| :cat:                         | &#128049;                               |  |

# Variantes

La syntaxe d'origine n'a jamais évolué et n'a pas été formellement standardisée.

Quelques variantes (*flavors*) sont donc apparues pour répondre à des besoins spécifiques (ex. tableaux avancés, notes de bas de page, formules mathématiques, etc.)

→ même socle commun mais fonctionnalités supplémentaires selon les variantes

Variantes les plus célèbres :

- CommonMark
- GitHub Flavored Markdown
- MultiMarkdown
- Pandoc's Markdown

→ Spécification conçue pour un usage universitaire, basée sur `pandoc` (convertisseur de documents) et implémentée dans R



# Éditeurs Markdown

Grande variété d'éditeurs Markdown !

Trois propositions :

- Éditeur Markdown à télécharger : **Zettlr**
- Éditeur Markdown en ligne : **HedgeDoc**
- Environnement de développement intégré (IDE) : **VS Codium**

Éditeur Markdown intégré dans divers outils :

- **Obsidian** : pour la prise de notes
- **Stylo** : pour l'édition sémantique de textes scientifiques
- **Rstudio** : pour la rédaction de documents interactifs
- **Jupyter Notebook** : pour la conception de notebooks
- **GitLab** : pour la publication de la documentation d'un projet



# Zettlr

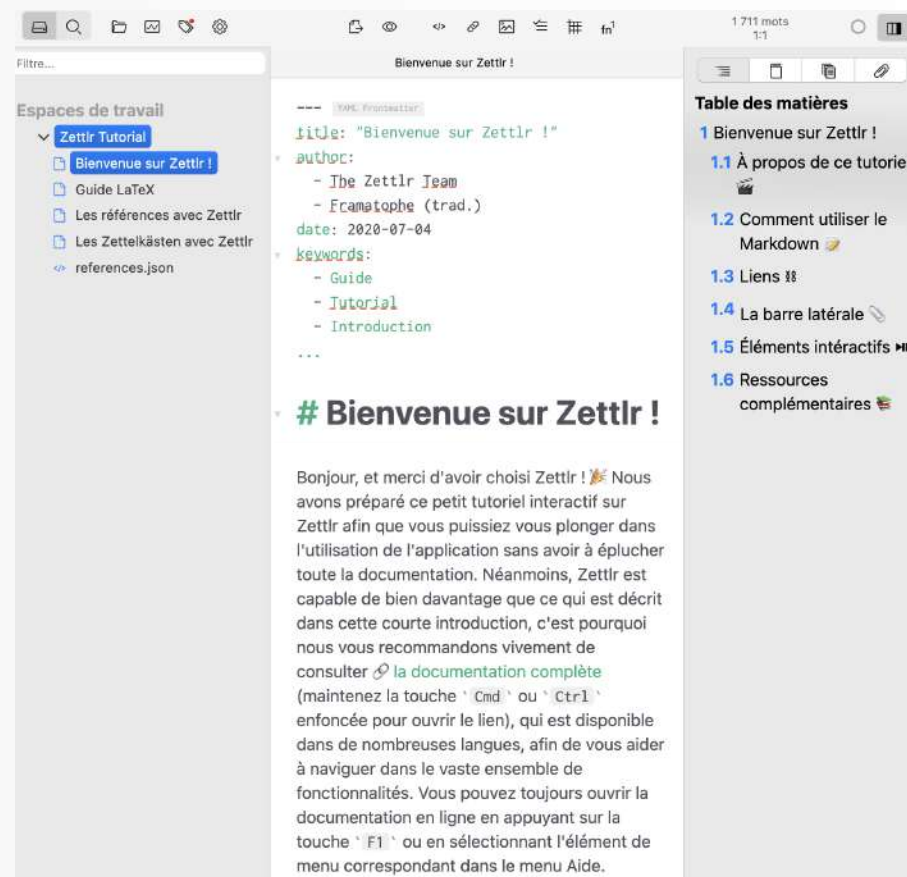
Éditeur de texte Markdown *open source* et multiplateforme,

→ Implémente d'autres solutions libres (Pandoc, Reveal.js, Zotero)

→ Particulièrement adapté aux usages académiques

## Principales fonctionnalités :

- Interface épurée et mode sans distraction
- Navigation simultanée et vue arborescente de plusieurs espaces de travail
- Gestion des références et notes de bas de page
- Vérification orthographique et correction automatique
- Nombreuses possibilités d'export personnalisables (PDF, HTML, DOCX...)
- Minuteur Pomodoro intégré (gestion du temps)



# HedgeDoc

Outil de prise de notes collaborative et simultanée (ex. CodiMD, basé sur HackMD)

<https://demo.hedgedoc.org>

## Principales fonctionnalités :

- Espaces de travail : lecture, écriture, les deux
- Thèmes : jour ou nuit
- Barre d'aide à la rédaction (et bouton aide)
- Aperçu des personnes connectées
- Gestion fine des permissions
- Historique des versions
- Page de visualisation partageable
- Mode présentation (avec `reveal.js`)
- Exports en MD et HTML

The screenshot displays the HedgeDoc interface. On the left is the code editor with the following content:

```

---
title: Les bases de la syntaxe Markdown
author: Daphné Mathelier (MSH Mondes)
date: 2023-01-29
---
# Un titre (titre de niveau 1)

## Un sous-titre (titre de niveau 2)

### Une section (titre de niveau 3)

On peut formater du texte en gras, en italique ou les deux !

Une liste non numérotée :
- élément 1
- élément 2
- élément 3

Une liste numérotée :
1. élément 1
2. élément 2
3. élément 3

Pour changer de paragraphe il faut laisser une une ligne blanche.

On peut mettre des commentaires avec entre les balises <!-- et --> et des hyperliens : [passage cliquable](URL)

On peut mettre 'des extraits de code' sur un ligne ou dans un bloc de code <!-- chunk --> :



```
python
print("toto")

```


```

On the right is the live preview, which shows the rendered document with the following structure:

- Un titre (titre de niveau 1)
- Un sous-titre (titre de niveau 2)
- Une section (titre de niveau 3)
- On peut formater du texte **en gras**, en *italique* ou ***les deux*** !
- Une liste non numérotée :
  - élément 1
  - élément 2
  - élément 3
- Une liste numérotée :
  - élément 1
  - élément 2
  - élément 3
- Pour changer de paragraphe il faut laisser une une ligne blanche.
- On peut mettre des commentaires avec entre les balises <!-- et --> et des hyperliens : [passage cliquable](#)
- On peut mettre des extraits de code sur un ligne ou dans un bloc de code :
 

```
print("toto")
```

# VSCodium

Environnement de développement intégré (**IDE**) multiplateforme

→ Version libre (sans collecte de données) du logiciel VS Code

## Principales fonctionnalités :

- Lecture et écriture dans tous les langages de programmation, voire en les cumulant !
- Panneau de prévisualisation en temps réel
- Explorateur de fichiers avec outil de recherche puissant
- Aide à la rédaction avec la coloration syntaxique et l'autocomplétion
- Console intégrée permettant notamment la gestion de version avec Git
- Beaucoup d'extensions comme Marp, Jupyter notebook, GitLab, reveal.js

The screenshot shows the VSCodium editor interface. The main editor displays a markdown file titled "20221207-Atelier-Git.md". The content includes a header with title, author, and date, followed by a table of contents and a list of steps for setting up Git. The preview pane on the right shows the rendered markdown, including a table of contents and a list of steps. The status bar at the bottom indicates the current file is a slide, line 7, column 1, in UTF-8 encoding, and is a markdown file.

```

1 ---
2 title: "20221207-Atelier-Git"
3 author: Brian Chauvel et Daphné Mathelier (MSH Mondes)
4 date: 2022-12-07
5 ---
6
7 Semaine DataSHS 2022 – Atelier Git
8
9 ## Étapes
10 1. Ouvrir son espace Sharedocs
11 2. L'installer en local avec Nextcloud
12 3. Installer Git sur son ordinateur
13 4. Créer le répertoire (repo) qui sera synchronisé avec RStudio et GitLab
14 5. Le synchroniser
15   - Depuis le serveur distant (GitLab → ordi)
16   - Vers le serveur distant (ordi → GitLab)
17
18 ## Rappels des commandes
19 ## Terminal
20 - 'ls' : liste du contenu d'un dossier
21 - 'mkdir' : créer un dossier
22 - 'cd' : changer de dossier ('cd ..' pour remonter)
23 - 'tab' : autocomplétion tab,
24 - '↑' (flèche du haut) : répète la commande précédente
25
26 ## Git
27 - 'git --version'
28 - 'git add' <!-- enregistrement d'une modif -->
29 - 'git commit -m "Message du commit"' <!-- ajout modif + explication -->
30 - 'git log' <!-- historique des actions effectuées -->
31 - 'git push' <!-- envoi des modifs au serveur distant -->
32 - 'git pull' <!-- récupération des modifs distantes -->
33
34 <!-- 'fetch' : informe sur les modifs distantes mais n'importe pas
35 - 'fork' (gitlab) créé une copie totalement indépendante du repo
36 - 'clone' (git) créé une copie reliée/synchronisée avec le repo -->
37
38 ## Étape 3 – Installation de git
39 Doc officielle : https://git-scm.com/book/fr/v2/D%C3%A9marrage-rapide-Installation-de-Git
40

```

## ShareDocs

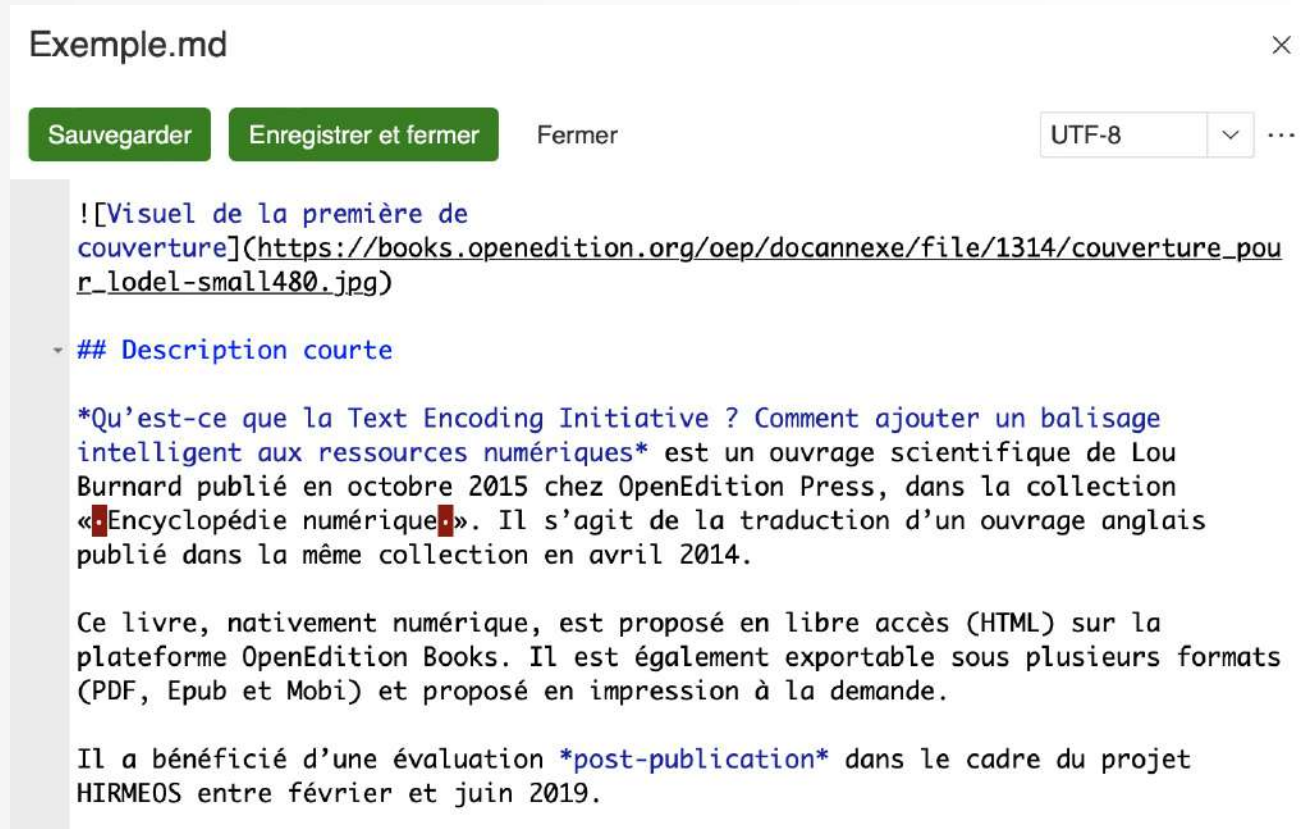
Solution de stockage en ligne  
(et en local) de l'IR\* **Huma-Num**

→ Accessible à partir du portail  
**Humanid** aux universitaires SHS

→ Comprend un éditeur  
Markdown intégré

*(C'est le cas dans la plupart des  
solutions de stockage en ligne  
puisque le .md est un format  
texte ouvert)*

Pas de fonctionnalités (exceptées la lecture et l'écriture) mais cela peut dépanner !



```
Exemple.md
[Visuel de la première de
couverture](https://books.openedition.org/oep/docannexe/file/1314/couverture_pou
r_lodel-small480.jpg)

## Description courte

*Qu'est-ce que la Text Encoding Initiative ? Comment ajouter un balisage
intelligent aux ressources numériques* est un ouvrage scientifique de Lou
Burnard publié en octobre 2015 chez OpenEdition Press, dans la collection
«Encyclopédie numérique». Il s'agit de la traduction d'un ouvrage anglais
publié dans la même collection en avril 2014.

Ce livre, nativement numérique, est proposé en libre accès (HTML) sur la
plateforme OpenEdition Books. Il est également exportable sous plusieurs formats
(PDF, Epub et Mobi) et proposé en impression à la demande.

Il a bénéficié d'une évaluation *post-publication* dans le cadre du projet
HIRMEOS entre février et juin 2019.
```

## 3 – Exemples d'utilisation

Un seul langage pour diverses utilisations !

Exemples d'utilisation :

- **Prendre des notes** → Obsidian
- **Réaliser une présentation** → Marp | reveal.js
- **Écrire un texte scientifique** → Stylo
- **Rédiger un document interactif** → R Markdown
- **Concevoir un notebook** → Jupyter Notebook
- **Documenter un projet** → Gitlab



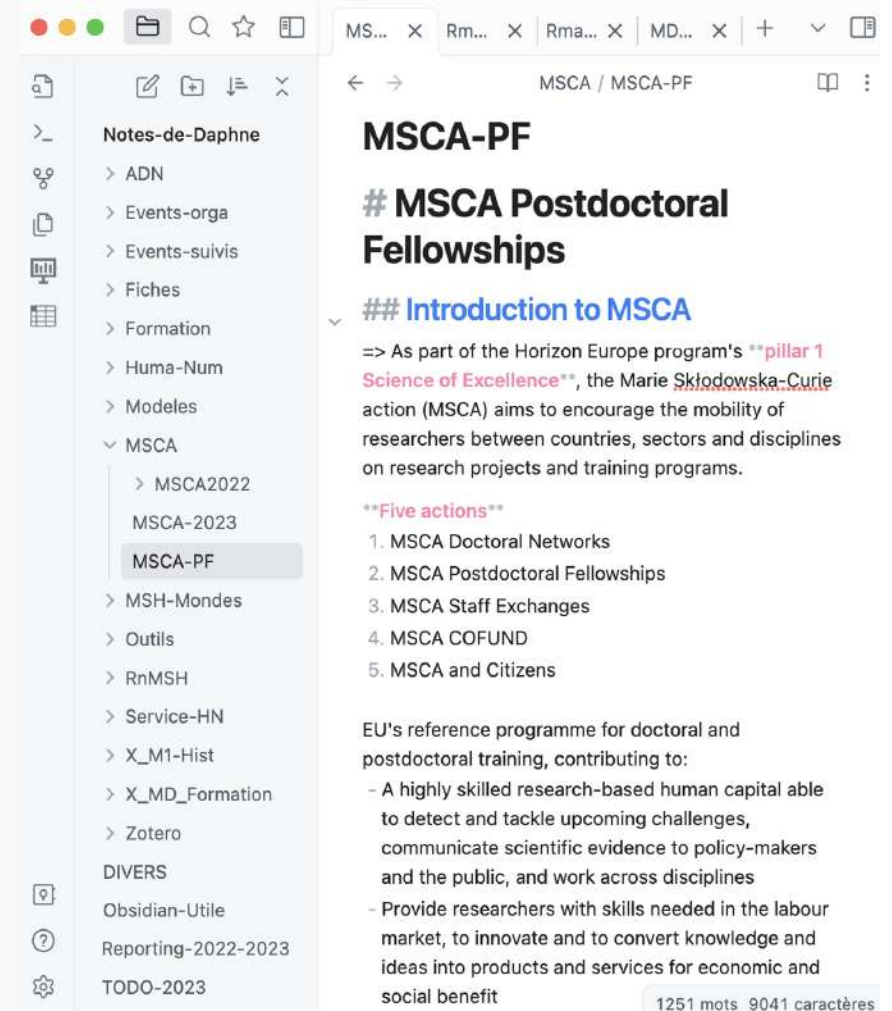
## Prendre des notes avec Obsidian

Application de prise de notes lancée en 2020

- Gratuite avec fonctionnalités payantes (**Sync** et **Publish**)
- Notes au format `md` stockées en local dans un coffre-fort

### Avantages

- **Ceux du Markdown** → simplicité et pérennité des fichiers
- **Centralisation** → toutes ses notes au même endroit, dans un ou plusieurs coffres-forts dont l'arborescence est visible
- **Synchronisation multisupport** : via un cloud (ou Sync)
- **Gestion de projet** → TO DO listes, notes journalières, calendrier intégré , etc.
- **Communauté active** → nombre croissant d'extensions et de thèmes (voir ceux d'**Awesome-obsidian**)
- **Personnalisation** → modèles, raccourcis, styles CSS, etc.



## Fonctionnalités

**Visualisation** : mode source ou aperçu en direct

**Liens internes** `[[ ]]` : `[[Nom-note]]` :

- Connexion des notes entre elles
- Création d'index ou de **vues graphiques**

**Mots clés** (#) : `#Mot-clé` ou `#Mot-clé/Mot-clé`

- Connexion des notes à des idées
- Plusieurs niveaux de mots clés
- Dans l'entête pour une bonne indexation

**Recherche** : par titre, mot clé, chemin d'accès, plein texte...

**Formats d'exports** :

- Par défaut : PDF + vue HTML et présentation
- Avec Pandoc : HTML, ePub, LaTeX, reveal.js/PPT, ODT/DOCX, et bien d'autres !

The screenshot shows a software interface with a file tree on the left and search options on the right.

**File Tree (Left):**

- ▼ Huma-Num
  - ▼ Formations
    - Atelier-20220407
    - Atelier-20220602
    - Atelier-20221014
  - Huma-Num
  - Huma-Num\_ANF-2021
  - Huma-Num\_ANF-2022
  - Huma-Num\_GT-2021-2022

**Search Options (Right):**

**## Formations**

- `[[Huma-Num/Huma-Num_ANF-2021]]`
- `[[Atelier-20220407]]`
- `[[Atelier-20220602]]`
- `[[Atelier-20221014]]` #À-compléter
- `[[Huma-Num/Huma-Num_ANF-2022]]`

**## Réseau des correspondants**

Voir `[[Huma-Num_GT-2021-2022]]`

**Options de recherche** ⓘ

- path:** correspondance avec le chemin du fichier
- file:** correspondance avec le nom du fichier
- tag:** recherche par mots-clés
- line:()** recherche par mots-clés
- section:()** recherche par mots-clés sous le même titre



MD\_TutoMate-txt

X\_MD\_Formation / MD\_TutoMate-txt

**\*\*Programme\*\***

- Découvrir Markdown <!-- quoi -->
- Écrire en Markdown <!-- comment -->
- Exemples d'utilisation <!-- pourquoi faire -->
  - Prendre des notes &rarr; `Obsidian`
  - Réaliser une présentation &rarr; `Marp`
  - Écrire un texte scientifique &rarr; `Stylo`
  - Documenter un projet &rarr; `Gitlab`
  - Concevoir un notebook R &rarr; `Rstudio`
  - Concevoir un notebook Python &rarr; `Jupyter`
- Ressources

Vue graphique

MD\_TutoMate-txt

X\_MD\_Formation / MD\_TutoMate-txt

## MD\_TutoMate-txt

### Tuto@Mate – Markdown pour les SHS

**Daphné MATHELIER**  
Service Humanités numériques (MSH Mondes)  
Correspondante MSH de l'IR\* Huma-Num

Objectifs de la formation : vous faire découvrir le langage MD et sa syntaxe, quelques éditeurs MD et quelques exemples d'utilisation

**Programme**

- Découvrir Markdown
- Écrire en Markdown
- Exemples d'utilisation
  - Prendre des notes → `Obsidian`
  - Réaliser une présentation → `Marp`
  - Écrire un texte scientifique → `Stylo`
  - Documenter un projet → `Gitlab`
  - Concevoir un notebook R → `Rstudio`
  - Concevoir un notebook Python → `Jupyter`
- Ressources

### Découvrir Markdown

#### Qu'est-ce que Markdown ?

Markdown est un **langage de balisage léger** facile à lire et à écrire créée en 2004 par [John Gruber](#) (avec l'aide d'[Aaron Swartz](#) pour la syntaxe)

Tuto@Mate – Markdown pour les SHS

- Découvrir Markdown
    - Qu'est-ce que Markdown ?
    - Avantages du Markdown
    - Le format Markdown
    - Markdown comme alternative à Word !
    - En bref
  - 2 – Écrire en Markdown
    - Variantes
    - Éditeurs Markdown
      - Zettlr
      - HedgeDoc
      - VS Codium
      - ShareDocs
  - 3 – Exemples d'utilisation
    - Prendre des notes avec Obsidian
      - Description
      - Avantages
      - Utilisation
      - Obsidian et Zotero
      - Extensions
    - Réaliser une présentation avec Markdown
      - Slides
      - Marp
      - Reveal.js
    - Écrire un texte scientifique avec Stylo
      - Description
      - Utilisation
      - Environnement d'édition

## Obsidian et Zotero

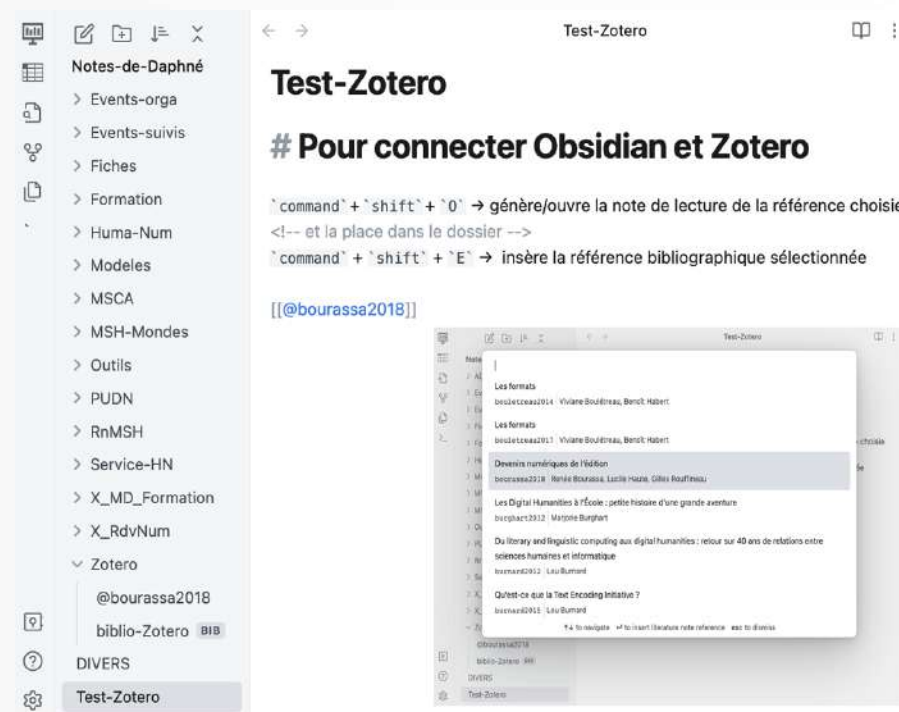
Pour citer des références d'une bibliothèque Zotero dans Obsidian → extension **Citation**

1. Installer le logiciel **Zotero** en local et l'extension **Better Bibtex**
2. Exporter sa bibliothèque Zotero au format **BibLaTeX** de préférence (ou **Better CSL JSON**)
3. Installer l'extension **Citation** dans Obsidian
4. Renseigner le chemin vers l'export **BibLaTeX** de sa bibliothèque Zotero
5. Renseigner le chemin vers le dossier où seront stockées les notes de lecture

`command + shift + E` → insère la référence sélectionnée

`command + shift + 0` → génère ou ouvre la note de lecture de la référence sélectionnée

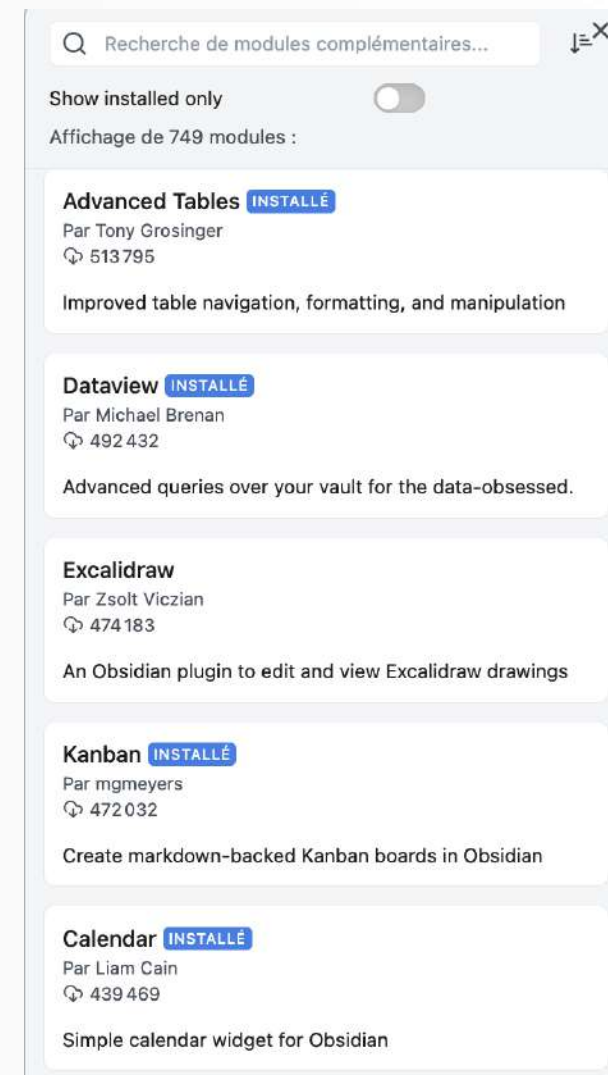
Il est aussi possible d'exporter ses notes de lecture Zotero dans Obsidian (plugin **Zotfile**)



# Extensions

Plus de 700 actuellement ! Petite sélection :

1. **Advanced Slides** → pour créer des présentations basées sur reveal.js
2. **Advanced Table** → pour faciliter la création/gestion des tableaux
3. **Better footnote** → pour faciliter la création/gestion des notes
4. **Calendar** → pour afficher un calendrier mensuel et créer des notes journalières ou des rapports hebdomadaires
5. **Citation** → pour rechercher et insérer des références bibliographiques issues de Zotero
6. **Kanban** → pour la gestion de projet selon la méthode Kanban
7. **Obsidian Pandoc** → pour exporter en n'importe quel format
8. **Pandoc Reference List** → pour voir les références bibliographiques citées dans un fichier
9. **Smart Typography** → pour convertir les symboles et la typographie anglaise en typographie française
10. **Tag Wrangler** → pour une gestion plus fine des mots clés



# Réaliser une présentation avec Markdown

**Exemples de trois solutions :** **1.** Fonctionnalité intégrée dans Obsidian ( *Slides* )  
**2.** Plugin *Marp* dans VSCode | **3.** Plugin *reveal.js* (dans Obsidian, Rstudio, Zettlr, VSCode...)

## 1. Slides

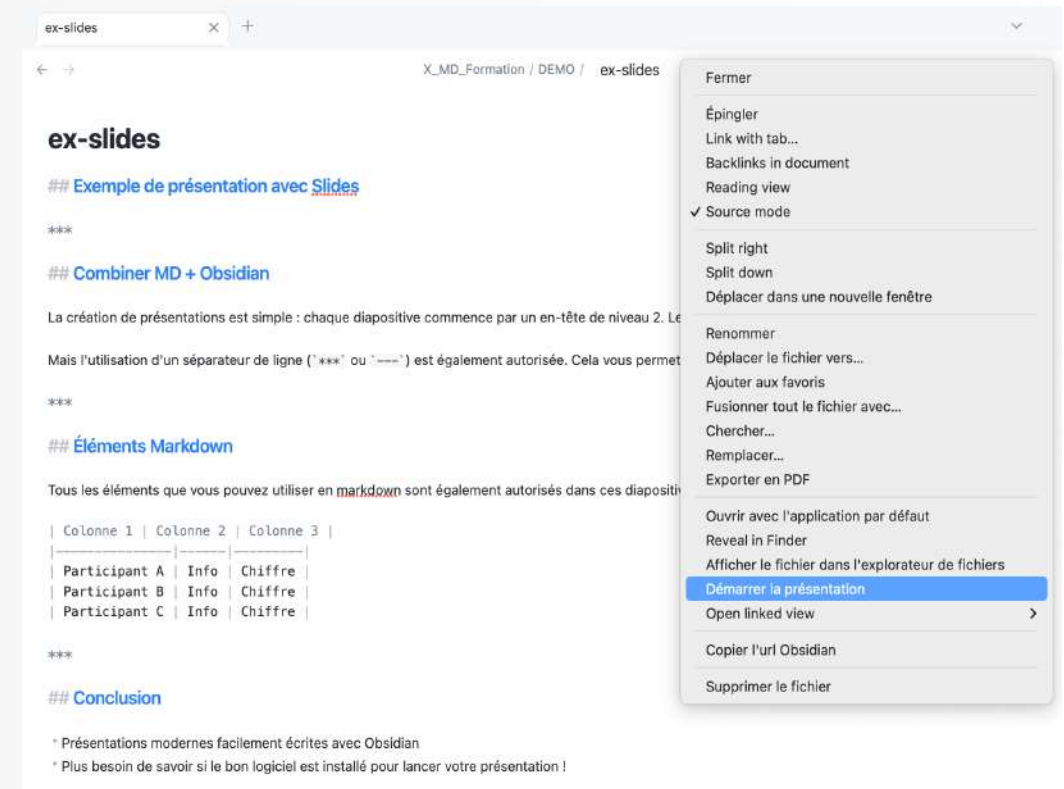
*Slides* : un des modules principaux d'Obsidian

Très simple à utiliser : écrire directement en MD en séparant les vignettes avec `---` ou `***` puis démarrer la présentation !

→ Inconvénient : pas de personnalisation ou de prise en compte des plugins

→ Alternatives dans Obsidian :

*Advanced Slides* et *Obsidian Pandoc*



## 2. Marp

Outil libre qui dispose d'une extension dans **VSCodium**

→ Permet de visualiser son support de présentation en temps réel et de l'exporter dans plusieurs formats

Éléments indispensables :

- marp: true dans l'entête
- --- pour séparer les vignettes

Personnalisation

- Choix entre plusieurs thèmes
- Divers paramètres dans l'entête
- Possibilité de charger son propre style CSS (paramétrage global ou par diapositive)

The image shows a side-by-side comparison of a presentation slide in its source code (left) and its rendered preview (right).

**Source Code (Left):**

```

195 ---
196 ### How to address Open Science practices in your proposal?
197 See the useful tips in the [HE program guide](https://ec.europa.eu/info/funding-tenders/opportunities/docs/2021-2027/horizon/guidance/programme-guide_horizon_en.pdf) (p. 41-42) and the [Annotated Model Grant Agreement](https://ec.europa.eu/info/funding-tenders/opportunities/docs/2021-2027/common/guidance/aga_en.pdf) (p. 151-160) on the following aspects:
198
199 - **Early and open sharing**: *Explain how you will implement early and open sharing and for which part of your expected output.*
200 - **Research data management**: *Outline how data or other research outputs (except for publications) will be managed.*
201 - **Reproducibility of research outputs**: *Outline the measures planned in the project that tend to increase reproducibility.*
202 - **Open access**: *Explain how you will meet the open access requirements.* <!--That is submission and immediate open access to publications and open access to data (the latter with some exceptions and within the deadlines set in the DMP) through a trusted repository, and under open licenses-->
203 - **Open peer review**: *Prefer open evaluation where possible.* <!--Prefer open peer review for your publications over traditional peer review anytime it is possible -->
204 - **Citizen, civil society and end-user engagement**: *Where/if appropriate, explain how citizen, civil society and end-users will be implemented in your project.*
205
206 ---
207 ### Concretely, what to do in your project?
208 ![center width:850](images/50-HE-concretely.png)
209 <!-- A table to summarize all this. -->
210
211 Source: European Commission, [Open Science in Horizon Europe](https://eua.eu/component/attachments/attachments.html?task=attachment&id=3250)
212
213 ---
214 ## Research Data
215 *Research data is any information that has been collected, observed, generated or created to validate original research findings.* Source: [POSTER](https://www.fosteropenscience.eu/node/2132)
216
217 **Expectations of the European Commission**
218 1. Write a Data Management Plan (DMP) and regular updates.
219 2. Manage your data well (according to FAIR principles).
220 3. Deposit your data in a trusted repository as soon as possible.
221 4. Ensure open access to the data as much as possible.
222 <!-- Data will be made as open as possible, but may remain as closed as necessary, to preserve legitimate interests or limitations. -->
223
224 **Tips and resources**
225 - Document your data as you go along.
226 - Discuss with ArScAn members about the collective practices. <!-- naming and organization of folders, sharing, tools -->
227 - Identify and asks the team research support *(like us!)*. <!-- See contacts at the end of this presentation -->

```

**Rendered Preview (Right):**

The preview shows a slide titled "How to address Open Science practices in your proposal?". It contains the same text as the source code but rendered as a presentation slide. Below the text, there is a table titled "Concretely, what to do in your project?".

| Area   | Key  | Recommendations |
|--|--|-----------------|
| Early and open sharing of research             | Open access to publications, preprints, data, software, etc.                     | Recommended     |
| Research data management                       | Data management plan (DMP)   | Recommended     |
| Reproducibility of research outputs            | Submission of publications and access to publications to enhance reproducibility | Recommended     |
| Open access to research outputs                | Open access to publications, preprints, data, software, etc.                     | Recommended     |
| Open peer review                               | Open access to publications, preprints, data, software, etc.                     | Recommended     |
| Citizen, civil society and end-user engagement | Open access to publications, preprints, data, software, etc.                     | Recommended     |

Source: European Commission, [Open Science in Horizon Europe](https://eua.eu/component/attachments/attachments.html?task=attachment&id=3250)

**Research Data**

Research data is any information that has been collected, observed, generated or created to validate original research findings. Source: [EDS116](https://www.fosteropenscience.eu/node/2132)

**Expectations of the European Commission**

1. Write a Data Management Plan (DMP) and regular updates.
2. Manage your data well (according to FAIR principles).
3. Deposit your data in a trusted repository as soon as possible.
4. Ensure open access to the data as much as possible.

**Tips and resources**

- Document your data as you go along.
- Discuss with ArScAn members about the collective practices.
- Identify and asks the team research support (like us!).

### 3. Reveal.js

Outil *open source* basé sur les technologies web (HTML, CSS, JS)

→ Présentations faciles à partager et qui fonctionnent partout

→ Beaucoup de thèmes et de possibilités de personnalisation

--- : pour séparer les vignettes (ou \*\*\* selon les extensions)

yam1 : pour configurer les paramètres de la présentation

Intégré dans de nombreux logiciels :

1. **Zettlr** : un des formats d'export depuis la version 0.19.0
2. **Obsidian** : plugins *Advanced Slides* et *Obsidian Pandoc*
3. **VSCodium** : plugin *vscode-reveal*
4. **Rstudio** : plugin *revealjs*

```
Fichiers-demo > ex-VSC-revealjs.md > ## Éléments Markdown
1  ✓ ## Ex de présentation avec reveal.js
2
3  ---
4
5  ✓ ## Combiner MD + VSCode + Reveal.js
6  La création de présentations à l'aide de est
   simple : chaque diapositive commence par un
   en-tête de niveau 2. Le texte qui suit devient le
   contenu de la diapositive.
7
8  Mais l'utilisation d'un séparateur de ligne
   (---) est également autorisée. Cela vous permet
   d'avoir plus d'espace sur votre diapositive.
9
10 <!-- note: note dissimulée -->
11
12 ---
13
14 ✓ ## Éléments Markdown
15 Tous les éléments que vous pouvez utiliser en
   markdown sont également autorisés dans ces
   diapositives.
16
17 | Colonne 1 | Colonne 2 | Colonne 3 |
18 |-----|-----|-----|
19 | Participant A | Info | Chiffre |
20 | Participant B | Info | Chiffre |
21 | Participant C | Info | Chiffre |
22
23 ---
24
25 ✓ ## Paramètres reveal.js
26 Il est possible de personnaliser la mise en forme
   avec du code CSS :
27
28 ✓ ```css
29 body {
30   background-color: rgb(40, 40, 40);
31   color: rgb(200, 200, 220);
32 }
33 ```
```

**1**

**Exemple de Pandoc (export reveal.js)**

La création de présentations à l'aide de Zettlr est simple : chaque diapositive commence par un en-tête de niveau 2. Le texte qui suit devient le contenu de la diapositive.

Mais l'utilisation d'un séparateur de ligne ("---" ou "---") est également autorisée. Cela vous permet d'avoir plus d'espace sur votre diapositive.

**## Éléments Markdown**

Tous les éléments que vous pouvez utiliser en markdown sont également autorisés dans ces diapositives.

| Colonne 1     | Colonne 2 | Colonne 3 |
|---------------|-----------|-----------|
| Participant A | Info      | Chiffre   |
| Participant B | Info      | Chiffre   |
| Participant C | Info      | Chiffre   |

**## Paramètres `reveal.js`**

Il est possible de personnaliser la mise en forme avec du code CSS :

```

css
body {
  background-color:rgb(40, 40, 40);
  color: rgb(200, 200, 220);
}

```

**2**

**ex-advanced-slides**

**## Exemple de présentation avec Advanced Slides**

**## Combiner MD + Advanced Slides (Obsidian)**

La création de présentations est simple : chaque diapositive commence par un en-tête de niveau 2. Le texte qui suit devient le contenu de la diapositive.

L'utilisation d'un séparateur de ligne ("---") est également autorisée. Cela vous permet d'avoir plus d'espace sur votre diapositive.

**## Éléments Markdown**

Tous les éléments que vous pouvez utiliser en markdown sont également autorisés dans ces diapositives.

| Colonne 1     | Colonne 2 | Colonne 3 |
|---------------|-----------|-----------|
| Participant A | Info      | Chiffre   |
| Participant B | Info      | Chiffre   |
| Participant C | Info      | Chiffre   |

**## Conclusion**

- Présentations modernes facilement écrites !
- Plus besoin de savoir si le bon logiciel est installé pour lancer votre présentation !

**3**

**## Exemple de présentation avec Pandoc (export reveal.js)**

La création de présentations à l'aide de est simple : chaque diapositive commence par un en-tête de niveau 2. Le texte qui suit devient le contenu de la diapositive.

Mais l'utilisation d'un séparateur de ligne ("---") est également autorisée. Cela vous permet d'avoir plus d'espace sur votre diapositive.

**## Éléments Markdown**

Tous les éléments que vous pouvez utiliser en markdown sont également autorisés dans ces diapositives.

| Colonne 1     | Colonne 2 | Colonne 3 |
|---------------|-----------|-----------|
| Participant A | Info      | Chiffre   |
| Participant B | Info      | Chiffre   |
| Participant C | Info      | Chiffre   |

**## Paramètres `reveal.js`**

Il est possible de personnaliser la mise en forme avec du code CSS :

```

css
body {
  background-color:rgb(40, 40, 40);
  color: rgb(200, 200, 220);
}

```

**4**

**R Markdown**

Il s'agit d'une présentation R Markdown. Markdown est une syntaxe de formatage simple pour la création de documents HTML, PDF et DOCX. Pour plus de détails sur l'utilisation de R Markdown, voir <http://rmarkdown.rstudio.com>.

Lorsque vous cliquez sur le bouton Knit, un document sera généré qui comprendra à la fois le contenu ainsi que la sortie de tous les morceaux de code R intégrés dans le document.

```

---
title: "reveal.js"
output:
  revealjs: revealjs_presentation:
  self_contained: false
  reveal_plugins:
    - search
    - zoom
    - notes
    - chalkboard
    - menu
  reveal_options:
    - chalkboard
  theme: chalkboard
  loggaleNotesButton: true
  menu:
  - numbers: true
---
[r_setup, include=FALSE]
knitr::opts_chunk$set(echo = FALSE)
---
## R Markdown
Il s'agit d'une présentation R Markdown. Markdown est une syntaxe de formatage simple pour la création de documents HTML, PDF et DOCX. Pour plus de détails sur l'utilisation de R Markdown, voir <http://rmarkdown.rstudio.com>.
Lorsque vous cliquez sur le bouton **Knit**, un document sera généré qui comprendra à la fois le contenu ainsi que la sortie de tous les morceaux de code R intégrés dans le document.
---
## Slide avec liste à puces
- Entrée 1
- Entrée 2
- Entrée 3
---
aside class="notes">

```

# Écrire un texte scientifique avec Stylo

Éditeur de texte sémantique WYSIWYM (*what you see is what you mean*) → Développé par la **CRCEN** depuis 2016 et intégré dans le portail **Humanid** depuis 2020

## Utilisation

- Outil de rédaction et d'édition en ligne, d'abord pensé pour la rédaction et la gestion d'articles scientifiques
- Également adapté et utilisé pour d'autres docs textuels (billets, rapports, mémoires...)
- Pensé pour les chercheur·es, les étudiant·es, les professionnel·les de l'édition et de l'informations scientifiques (différents exports, métadonnées, gestion des versions...)
- Reposant sur des standards ouverts et une triple structuration des contenus : métadonnées, texte, bibliographie

The screenshot shows the Stylo documentation page with the following content:

**Stylo** Daphné Mathelier Books Articles Log out ↗ © Documentation

**## Introduction**

Stylo est un éditeur de texte scientifique. Pour faire vos premiers pas sur Stylo, commencez par éditer cet article.

Stylo utilise le format *markdown* pour baliser et styler le texte. Cet article présente la syntaxe de base du markdown, et une documentation plus complète [lest accessible ici](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) (https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet).

Vous pouvez visualiser l'article à tout moment en cliquant sur le bouton **Preview** dans le menu de gauche.

**## Les titres**

Une modification

Les titres de niveaux 2 doivent être balisés avec 2 `#` (`##`) et non un seul, car le titre de niveau 1 correspond au titre de l'article, déclaré dans les métadonnées.

**### Titres de niveau 3**

Les titres de niveaux 3 doivent être balisés avec 3 `#` et ainsi de suite.

Un saut de ligne correspond au début d'un nouveau paragraphe.

**## Syntaxe minimale**

**### Gras et italique**

Voici du texte en *italique*. Voici du texte en **gras**.

**### Commentaire**

La ligne ci-dessous n'apparaîtra pas dans le document final.

`<!-- Cette ligne sera traitée comme un commentaire, elle n'apparaîtra pas dans le document final -->`

**### Images**

On peut insérer des images:

`![[Titre de mon image](https://avatars2.githubusercontent.com/u/16691667?s=200&v=4)]`

Notez que le «Titre de mon image» sera pris en compte comme légende de l'image dans l'article.

**### Listes**

Les listes non numérotées:

- item
- item



# Environnement d'édition

## Trois modules d'édition :

- Gestion éditoriale (à gauche) : versions, sommaire, bibliographie, statistiques
- Espace d'écriture (au centre)
- Métadonnées (à droite)

Article **How to Stylo** : présent dans chaque compte Stylo pour aider à utiliser l'outil  
 → Page de **visualisation**

# Environnement de visualisation

Trois volets : Table des matières | Métadonnées et corps du texte | Annotation

The screenshot shows the Stylo editor interface with three main panels:

- Versions (Left Panel):** A list of document versions with columns for version number, author, date, and actions (Compare, Export, Preview). The current version is BAT (3.0) by Emma, dated 19/06/2019.
- Table of Contents (Middle Panel):** A list of document sections with their corresponding page numbers. Sections include Introduction, Lumières orientales, Les anciens cadres d'Orient, Les relats orientaux, and Les mouvements indépendantistes.
- Metadata (Right Panel):** A form for entering document metadata, including Title, Subtitle, Date, Language, License, and Author information.

The main text area in the center contains the article content, which is partially obscured by large, semi-transparent text overlays: "Gestionnaire de versions", "Gestionnaire de métadonnées", "Table des matières", "Gestionnaire de bibliographie", and "Statistiques".

**How to Stylo** close

by **Daphné Mathelier**  
Last saved 2 days ago

> **Versions**

✓ **Table of contents**

- Introduction
- Les titres
  - ↳ Titres de niveau 3
- Syntaxe minimale
  - ↳ Gras et italique
  - ↳ Commentaire
  - ↳ Images
  - ↳ Listes
- Appareil critique
  - ↳ Notes de bas de page
  - ↳ Les références
  - ↳ Les citations
- Versions et export
  - ↳ Métadonnées
  - ↳ Preview et annotation
  - ↳ Export
- Annotations sémantiques
  - ↳ Annotation dans un paragraphe
  - ↳ Annotation de plusieurs paragraphes
- Bibliographie

✓ **Bibliography**

Jack, Goody | La Raison graphique. La do... 1979  
@goody\_raison\_1979

✓ **Stats**

**\_\_Stylo\_\_** close Daphné Mathelier Books Articles Log out (-) Documentation

**## Introduction**

Stylo est un éditeur de texte scientifique. Pour faire vos premiers pas sur Stylo, commencez par éditer cet article.

Stylo utilise le format *\*markdown\** pour baliser et styler le texte. Cet article présente la syntaxe de base du markdown, et une documentation plus complète [\[est accessible ici\]](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) (<https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>).

Vous pouvez visualiser l'article à tout moment en cliquant sur le bouton **\*\*Preview\*\*** dans le menu de gauche.

**## Les titres**

Les titres de niveaux 2 doivent être balisés avec 2 '#' ('##') et non un seul, car le titre de niveau 1 correspond au titre de l'article, déclaré dans les métadonnées.

**### Titres de niveau 3**

Les titres de niveaux 3 doivent être balisés avec 3 '#' et ainsi de suite.

Un saut de ligne correspond au début d'un nouveau paragraphe.

**## Syntaxe minimale**

**### Gras et italique**

Voici du texte en *\*italique\**. Voici du texte en **\*\*gras\*\***.

**### Commentaire**

La ligne ci-dessous n'apparaîtra pas dans le document final.

```
<!-- Cette ligne sera traitée comme un commentaire, elle n'apparaîtra pas dans le document final -->
```

**### Images**

On peut insérer des images:

`![[Titre de mon image]](https://avatars2.githubusercontent.com/u/16691667?s=200&v=4)`

Notez que le «Titre de mon image» sera pris en compte comme légende de l'image dans l'article.

**### Listes**

Les listes non numérotées:

- item
- item
- item

Les listes numérotées:

1. item
2. item
3. item

L'ordre des chiffres n'est pas important:

1. item
2. item

**Metadata** close

Basic Mode | Editor Mode | Raw Mode

Title\*

Subtitle

Date

Language

License

**Bibliography**

Display

**Authors**

Last name

First name

ORCID

VIAF

FOAF

ISNI

Wikidata

Last name

First name

ORCID

# Rédiger un document reproductible avec R Markdown

`rmd` : format et package développé par **RStudio** (mais qui peut s'ouvrir dans d'autres IDE)

→ Permet de réunir dans un seul document du texte, du code et le résultat de son exécution

- Mise à disposition et suivi des méthodes
- Inscription dans la tradition de la programmation lettrée et dans une démarche de reproductibilité

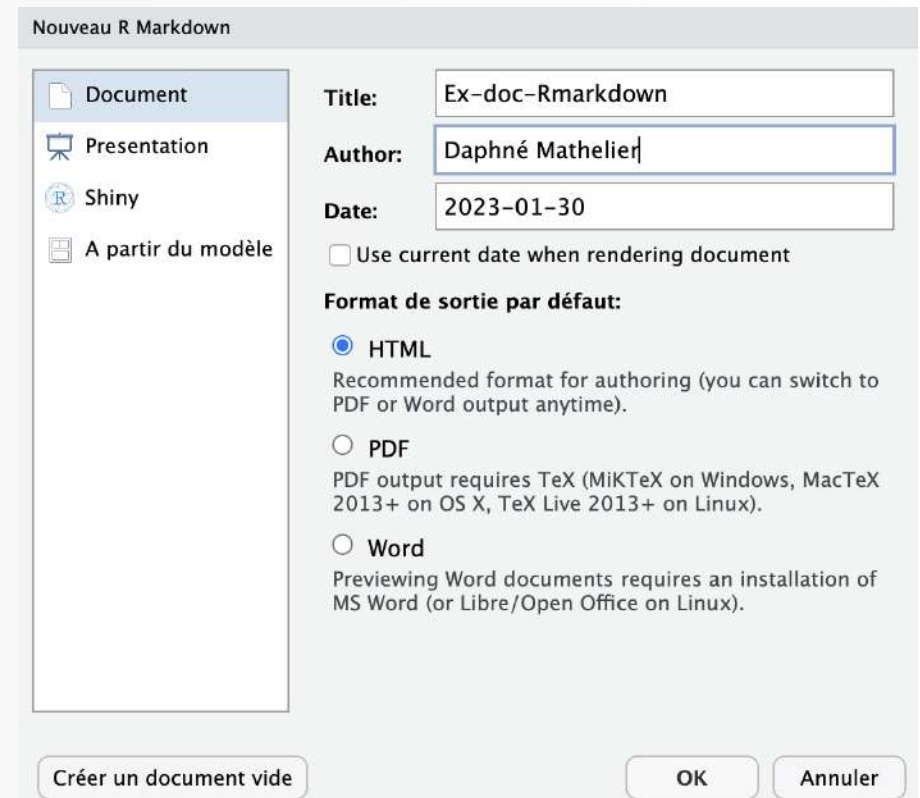
Package installé par défaut dans RStudio

ou à lancer dans la console dans un autre IDE :

```
install.packages ('rmarkdown', dependencies = TRUE)
```

Ressources utiles :

- R Markdown, [Authoring quick tour](#), [Cheatsheet](#), [Cookbook](#), [Reference Guide](#)
- Bibliothèque de l'Université de Montréal (2021), [Atelier R Markdown](#), [présentation](#) et [captation](#)



## Trois composantes d'un fichier R Markdown

1. Les métadonnées dans l'entête au format `yaml`
2. Le corps du texte au format `md`
3. Les blocs de code (*chunk*) au format `R`

|  |                                       |
|--|---------------------------------------|
| <pre>1 --- 2 title: "untitled" 3 output: html_document 4 ---</pre>   | Entête YAML                           |
| <pre>5 6 ```{r setup, include=FALSE} 7 knitr::opts_chunk\$set(echo = TRUE) 8 ```</pre>   | Options globales<br>des blocs de code |
| <pre>9 10 ## R Markdown 11 12 This is an R Markdown document. Markdown is a simple formatting syntax for 13 authoring HTML, PDF, and MS word documents. For more details on using R 14 Markdown see &lt;<a href="http://rmarkdown.rstudio.com">http://rmarkdown.rstudio.com</a>&gt;. 15 16 when you click the <b>knit</b> button a document will be generated that includes 17 both content as well as the output of any embedded R code chunks within the 18 document. You can embed an R code chunk like this:</pre> | Texte balisé<br>en markdown           |
| <pre>19 20 ```{r cars} 21 summary(cars) 22 ```</pre>   | Blocs de code                         |

## Fonctionnalités

- Gestion et personnalisation des métadonnées
- Rédaction du texte selon la syntaxe Markdown
- Paramétrage et exécution du code R
- Création de graphiques et de tableaux
- Gestion des références bibliographies `bibtex`
- Création de ses propres modèles de document
- Combinaison possible avec d'autres packages comme `rticle`, `reveal.js`, `bookdown`
- Formats de sorties : au départ HTML, PDF, DOCX et désormais beaucoup d'autres !
- Options de partage en ligne avec **RPubs**

**Et aussi : Notebook R** : format d'export interactif

**Quarto** : système de publication basé sur `md` et `pandoc`

```

1 ---
2 title: "Ex-doc-Rmarkdown"
3 author: "Daphné Mathelier"
4 date: "2023-01-30"
5 output: html_document
6 ---
7
8 <!-- paramètres généraux des blocs de code (chunk)-->
9 ```{r setup, include=FALSE}
10 knitr::opts_chunk$set(echo = TRUE)
11 ```
12
13 ## Rédiger du texte au format `md`
14 Ceci est un document R Markdown.
15
16 Markdown est une syntaxe de formatage simple pour la création de documents
17 HTML, PDF et
18 DOCX.
19
20 Pour plus de détails sur l'utilisation de R Markdown, voir
21 [http://rmarkdown.rstudio.com](http://rmarkdown.rstudio.com).
22
23 Lorsque vous cliquez sur le bouton Tricoter (*knit*), un document est
24 généré et inclut à la fois le contenu ainsi que la sortie de tout bloc de code
25 R (*chunk*) intégré dans le document.
26
27 <!-- bouton knit = fonction render() -->
28
29 ## Intégrer un bloc de code R
30
31 ```{r cars}
32 summary(cars)
33 ```
34
35 ## Intégrer un graphique
36
37 ```{r pressure, echo=FALSE, fig.height=3, fig.width=4}
38 plot(pressure)
39 ```
40
41 Notez que le paramètre `echo = FALSE` a été ajouté au morceau de code pour
42 empêcher l'impression du code R qui a généré le graphique.

```

```

1 ---
2 title: "Ex-doc-Rmarkdown"
3 author: "Daphné Mathelier"
4 date: "2023-01-30"
5 output: html_document
6 ---
7
8 <!-- paramètres généraux des blocs de code (chunk)-->
9 ```{r setup, include=FALSE}
10 knitr::opts_chunk$set(echo = TRUE)
11 ```
12
13 ## Rédiger du texte au format `md`
14 Ceci est un document R Markdown.
15
16 Markdown est une syntaxe de formatage simple pour la création de documents
17 HTML, PDF et
18 DOCX.
19
20 Pour plus de détails sur l'utilisation de R Markdown, voir
21 [http://rmarkdown.rstudio.com](http://rmarkdown.rstudio.com).
22
23 Lorsque vous cliquez sur le bouton Tricoter (*knit*), un document est
24 généré et inclut à la fois le contenu ainsi que la sortie de tout bloc de code
25 R (*chunk*) intégré dans le document.
26
27 <!-- bouton knit = fonction render() -->
28
29 ## Intégrer un bloc de code R
30
31 ```{r cars}
32 summary(cars)
33 ```
34
35 ## Intégrer un graphique
36
37 ```{r pressure, echo=FALSE, fig.height=3, fig.width=4}
38 plot(pressure)
39 ```
40
41 Notez que le paramètre `echo = FALSE` a été ajouté au morceau de code pour
42 empêcher l'impression du code R qui a généré le graphique.

```

## 1. Fichier source rmarkdown

## Ex-doc-Rmarkdown

Daphné Mathelier  
2023-01-30

### Rédiger du texte au format md

Ceci est un document R Markdown.  
 Markdown est une syntaxe de formatage simple pour la création de documents HTML, PDF et DOCX.  
 Pour plus de détails sur l'utilisation de R Markdown, voir <http://rmarkdown.rstudio.com>.  
 Lorsque vous cliquez sur le bouton **Tricoter** (*knit*), un document est généré et inclut à la fois le contenu ainsi que la sortie de tout bloc de code R (*chunk*) intégré dans le document.

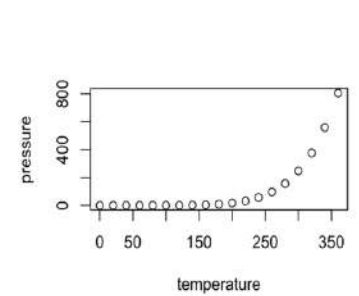
### Intégrer un bloc de code R

```
summary(cars)
```

| ##          | speed | dist           |
|-------------|-------|----------------|
| ## Min.     | : 4.0 | Min. : 2.00    |
| ## 1st Qu.: | :12.0 | 1st Qu.: 26.00 |
| ## Median : | :15.0 | Median : 36.00 |
| ## Mean :   | :15.4 | Mean : 42.98   |
| ## 3rd Qu.: | :19.0 | 3rd Qu.: 56.00 |
| ## Max. :   | :25.0 | Max. :120.00   |

## 2. Sortie HTML

### Intégrer un graphique



Notez que le paramètre `echo = FALSE` a été ajouté au morceau de code pour empêcher l'impression du code R qui a généré le graphique.

## Ex-doc-Rmarkdown

Daphné Mathelier  
2023-01-30

### Rédiger du texte au format md

Ceci est un document R Markdown.  
 Markdown est une syntaxe de formatage simple pour la création de documents HTML, PDF et DOCX.  
 Pour plus de détails sur l'utilisation de R Markdown, voir <http://rmarkdown.rstudio.com>.  
 Lorsque vous cliquez sur le bouton **Tricoter** (*knit*), un document est généré et inclut à la fois le contenu ainsi que la sortie de tout bloc de code R (*chunk*) intégré dans le document.

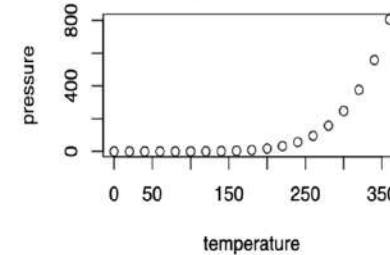
### Intégrer un bloc de code R

```
summary(cars)
```

| ##          | speed | dist           |
|-------------|-------|----------------|
| ## Min.     | : 4.0 | Min. : 2.00    |
| ## 1st Qu.: | :12.0 | 1st Qu.: 26.00 |
| ## Median : | :15.0 | Median : 36.00 |
| ## Mean :   | :15.4 | Mean : 42.98   |
| ## 3rd Qu.: | :19.0 | 3rd Qu.: 56.00 |
| ## Max. :   | :25.0 | Max. :120.00   |

## 3. Sortie PDF

### Intégrer un graphique



Notez que le paramètre `echo = FALSE` a été ajouté au morceau de code pour empêcher l'impression du code R qui a généré le graphique.

# rmarkdown :: CHEAT SHEET

## What is rmarkdown?

- .Rmd files** - Develop your code and ideas side-by-side in a single document. Run code as individual chunks or as an entire document.
- Dynamic Documents** - Knit together plots, tables, and results with narrative text. Render to a variety of formats like HTML, PDF, MS Word, or MS PowerPoint.
- Reproducible Research** - Upload, link to, or attach your report to share. Anyone can read or run your code to reproduce your work.

## Workflow

1. Open a new .Rmd file in the RStudio IDE by going to **File > New File > R Markdown**.
2. **Embed code** in chunks. Run code by line, by chunk, or all at once.
3. **Write text** and add tables, figures, images, and citations. Format with Markdown syntax or the RStudio Visual Markdown Editor.
4. **Set output format(s) and options** in the YAML header. Customize themes or add parameters to execute or add interactivity with Shiny.
5. **Save and render** the whole document. Knit periodically to preview your work as you write.
6. **Share your work**

## Embed Code with knitr

**CODE CHUNKS**  
Surround code chunks with ````r` and ````` or use the Insert Code Chunk button. Add a chunk label and/or chunk options inside the curly braces after `````.

```
```{r chunk="label", include=FALSE}
summary(mtcars)
```
```

### SET GLOBAL OPTIONS

Set options for the entire document in the first chunk.  
````{r, include=FALSE}
knitr::opts_chunk$set(message = FALSE)
````

### INLINE CODE

Insert `"r <code>"` into text sections. Code is evaluated at render and results appear as text.

```
"Built with r" getRversion() ~ "Built with 4.1.0"
```

## SOURCE EDITOR

## VISUAL EDITOR

OPTION	DEFAULT	EFFECTS
echo	TRUE	display code in output document
error	FALSE	TRUE (display error messages in doc) FALSE (stop render when error occurs)
eval	TRUE	run code in chunk
include	TRUE	include chunk in doc after knitting
message	TRUE	display code messages in document
warning	TRUE	display code warnings in document
results	"markup"	"hide" (don't display results) "hold" (put all results below all code) "self" ("self" or "center")
fig.align	"default"	alt text for figure
fig.alt	NULL	figure caption as a character string
fig.cap	NULL	prefix for generating figure file paths
fig.path	"figures"	plot dimensions in inches
fig.width & fig.height	7	rescales output width, e.g. "75%", "300px"
collapse	FALSE	collapse all sources & output into a single block
comment	"#"	prefix for each line of results
child	NULL	files() to knit and then include
pur1	TRUE	include or exclude a code chunk when extracting source code with knitr::purl()

See more options and defaults by running `str(knitr::opts_chunk$get())`

## RENDERED OUTPUT

## Insert Citations

Create citations from a bibliography file, a Zotero library, or from DOI references.

### BUILD YOUR BIBLIOGRAPHY

- Add BibTeX or CSL bibliographies to the YAML header.
- Add bibliography references .bib
- Link-citations: TRUE

### INSERT CITATIONS

- Access the **Insert Citations** dialog in the Visual Editor by clicking the **@** symbol in the toolbar or by clicking **Insert > Citation**.
- Add citations with markdown syntax by typing `[@cite]` or `@cite`.

## Insert Tables

Output data frames as tables using `kable(data, caption)`.

## Write with Markdown

The syntax on the left renders as the output on the right.

## More Header Options

**PARAMETERS**  
Parameterize your documents to reuse with new inputs (e.g., data, values, etc.).

## REUSABLE TEMPLATES

1. Create a new package with a `inst/mardown/` templates directory.
2. Add a folder containing `template.yaml` (below) and `skeleton.Rmd` (template contents).
3. Install the package to access template by going to **File > New R Markdown > From Template**.

## Set Output Formats and their Options in YAML

Use the document's YAML header to set an output format and customize it with output options.

```
title: "My Document"
author: "Author Name"
output:
  html_document:
    toc: TRUE
```

OUTPUT FORMAT	CREATES
html_document	.html
pdf_document	.pdf
word_document	Microsoft Word (.docx)
powerpoint_presentation	Microsoft Powerpoint (.pptx)
odt_document	OpenDocument Text
rtf_document	Rich Text Format
md_document	Markdown
github_document	Markdown for Github
ioslides_presentation	ioslides HTML slides
slidy_presentation	Slidy HTML slides
beamer_presentation	Beamer slides

\* Requires LaTeX, use `tinytex::install_tinytex()`  
 Also see `flexdashboard`, `bookdown`, `distill`, and `blogdown`.

IMPORTANT OPTIONS	DESCRIPTION	HTML	PDF	MS Word	MS PPT
anchor_sections	Show section anchors on mouse hover (TRUE or FALSE)	X			
citation_package	The LaTeX package to process citations ("default", "natbib", "biblatex")		X		
code_download	Give readers an option to download the .Rmd source code (TRUE or FALSE)	X			
code_folding	Let readers to toggle the display of R code ("none", "hide", or "show")	X			
css	CSS or SCSS file to use to style document (e.g. "style.css")	X			
dev	Graphics device to use for figure output (e.g. "png", "pdf")	X	X		
df_print	Method for printing data frames ("default", "kable", "tbl", "paged")	X	X	X	X
fig_caption	Should figures be rendered with captions (TRUE or FALSE)	X	X	X	X
highlight	Syntax highlighting ("tango", "pygments", "kate", "zenburn", "textmate")	X	X		
includes	File of content to place in doc ("in_header", "before_body", "after_body")	X	X		
keep_md	Keep the Markdown .md file generated by knitting (TRUE or FALSE)	X	X	X	X
keep_tex	Keep the intermediate TEX file used to convert to PDF (TRUE or FALSE)	X			
latex_engine	LaTeX engine for producing PDF output ("pdflatex", "xelatex", or "lualatex")	X			
reference_docx/_doc	doc/ppsx file containing styles to copy in the output (e.g. "file.docx", "file.pptx")	X	X		
theme	Theme options (see <code>Boothwatch</code> and <code>Custom Themes</code> below)	X			
toc	Add a table of contents at start of document (TRUE or FALSE)	X	X	X	X
toc_depth	The lowest level of headings to add to table of contents (e.g. 2, 3)	X	X	X	X
toc_float	Float the table of contents to the left of the main document content (TRUE or FALSE)	X			

Use `?output-format` to see all of a format's options, e.g. `?html_document`

## Render

- When you render a document, **Knit** does:
1. Runs the code and embeds results and text into an .md file with `knitr`.
  2. Converts the .md file into the output format with `Pandoc`.



**Save**, then **Knit** to preview the document output. The resulting HTML/PDF/MS Word/etc. document will be created and saved in the same directory as the .Rmd file.

Use `rmknitr::render()` to render/knit in the R console. See `?render` for available options.

## Share

- Publish on RStudio Connect** to share R Markdown documents securely, schedule automatic updates, and interact with parameters in real time. [rstudio.com/products/connect/](https://rstudio.com/products/connect/)



# Concevoir un notebook Python avec Jupyter Notebook

Interface de programmation interactive (en local ou dans l'application web)

- Fait partie du projet **Jupyter**, conçu initialement pour trois langages : Julia , Python , R
- Visualisation et partage facilité via le format `.ipynb`

→ Idéale pour apprendre à programmer, pour enseigner et pour documenter ses codes

Plusieurs manières d'accéder à Jupyter Notebook :

- Via le terminal (à l'aide de gestionnaires de paquets comme `conda` )
- Via la distribution **Anaconda** (application bureau)
- Via les extensions des IDE (ex. **l'extension** dans VSCode)

*Précision : rien n'est stocké par Jupyter, le serveur s'exécute en local sur notre machine donc pas besoin d'être connecté à Internet*





## Fonctionnalités

- Choix du langage à la création du notebook
- Deux modes de cellules : saisie (vert) et commande (bleu)
- Trois types de cellules : texte, code et Raw NBConvert
- Édition manuelle des métadonnées au format json
- Possibilité d'attribuer aux cellules des mots clés, pièces jointes ou caractéristiques de présentation
- Rédaction du texte avec la syntaxe et possibilités du md
- Écriture facilitée du code (coloration, complétion, indentation, numérotation des lignes...)
- Raccourcis (*magic commands* ; ex %whos et %history)
- Imports d'images et de graphiques
- Beaucoup de formats d'export (mais prévoir certaines installations au préalable comme LaTeX et pandoc)

The screenshot shows a Jupyter Notebook titled "Ex-notebook-Python". The interface includes a top bar with "Se déconnecter" and "Python 3 (ipykernel)". Below is a menu bar with options like "Fichier", "Édition", "Affichage", "Insérer", "Cellule", "Noyau", "Widgets", and "Aide". A toolbar contains icons for file operations and execution. The main content area has a heading "Exemple d'un notebook avec Python" and a sub-heading "Exemple d'exercice". The exercise text reads: "Une personne doit choisir sa boisson au distributeur d'une station-service :". A list of options is provided: 1. Café court non sucré, 2. Café court sucré, 3. Café long non sucré, 4. Café long sucré. A code cell contains a Python script that simulates this choice process. The output of the code cell shows the same list of options and the user's input: "Veuillez saisir votre choix de boisson : 3" and "Choix de boisson : Café long non sucré".

**Exemple d'un notebook avec Python**

**Exemple d'exercice**

Une personne doit choisir sa boisson au distributeur d'une station-service :

1. Café court non sucré
2. Café court sucré
3. Café long non sucré
4. Café long sucré

Écrire le programme qui simule cette situation en demandant de saisir son choix (1 à 4) et qui l'affiche ensuite.

```
Entrée [1]: print("\n1 : Café court non sucré \n2 : Café court sucré \n3 : Café long non sucré \n4 : Café long sucré\n")
choix=int (input('Veuillez saisir votre choix de boisson : '))
if choix ==1 :
    print("Choix de boisson : Café court non sucré\n")
if choix ==2 :
    print("Choix de boisson : Café court sucré\n")
if choix ==3 :
    print("Choix de boisson : Café long non sucré\n")
if choix ==4 :
    print("Choix de boisson : Café long sucré\n")
```

1 : Café court non sucré  
2 : Café court sucré  
3 : Café long non sucré  
4 : Café long sucré

Veuillez saisir votre choix de boisson : 3  
Choix de boisson : Café long non sucré

Entrée [ ]:

Ex-notebook-Python

Python 3 (ipykernel)

Fichier Édition Affichage Insérer Cellule Noyau Widgets Aide

Nouveau Notebook  
Ouvrir...

Faire une copie...  
Save as...  
Renommer...  
Créer une nouvelle sauvegarde [KS]  
Restaurer la sauvegarde

Imprimer l'aperçu  
Télécharger au format  
Faire confiance à ce Notebook  
Fermer et arrêter

Exercice

1. Fichier source ipynb

Entrée [1]:

```
print("\n1 : Café court non sucré\n2 : Café court sucré\n3 : Café long non sucré\n4 : Café long sucré")
choix=int(input("Veuillez saisir votre choix de boisson : "))
if choix ==1 :
    print("Choix de boisson : Café court non sucré\n")
if choix ==2 :
    print("Choix de boisson : Café court sucré\n")
if choix ==3 :
    print("Choix de boisson : Café long non sucré\n")
if choix ==4 :
    print("Choix de boisson : Café long sucré\n")
```

```
1 : Café court non sucré
2 : Café court sucré
3 : Café long non sucré
4 : Café long sucré
```

Choix de boisson : Café long non sucré

## Exemple d'un notebook avec Python

### Exemple d'exercice

Une personne doit choisir sa boisson au distributeur d'une station-service :

1. Café court non sucré
2. Café court sucré
3. Café long non sucré
4. Café long sucré

Écrire le programme qui simule cette situation en demandant de saisir son choix (1 à 4) et qui l'affiche ensuite.

```
In [1]: print("\n1 : Café court non sucré\n2 : Café court sucré\n3 : Café long non sucré\n4 : Café long sucré")
choix=int(input("Veuillez saisir votre choix de boisson : "))
if choix ==1 :
    print("Choix de boisson : Café court non sucré\n")
if choix ==2 :
    print("Choix de boisson : Café court sucré\n")
if choix ==3 :
    print("Choix de boisson : Café long non sucré\n")
if choix ==4 :
    print("Choix de boisson : Café long sucré\n")
```

```
1 : Café court non sucré
2 : Café court sucré
3 : Café long non sucré
4 : Café long sucré
```

Choix de boisson : Café long non sucré

### 2. Sortie HTML

## Ex-notebook-Python

February 1, 2023

### 1 Exemple d'un notebook avec Python

#### 1.1 Exemple d'exercice

Une personne doit choisir sa boisson au distributeur d'une station-service :

1. Café court non sucré
2. Café court sucré
3. Café long non sucré
4. Café long sucré

Écrire le programme qui simule cette situation en demandant de saisir son choix (1 à 4) et qui l'affiche ensuite.

```
[1]: print("\n1 : Café court non sucré\n2 : Café court sucré\n3 : Café long non sucré\n4 : Café long sucré")
choix=int(input("Veuillez saisir votre choix de boisson : "))
if choix ==1 :
    print("Choix de boisson : Café court non sucré\n")
if choix ==2 :
    print("Choix de boisson : Café court sucré\n")
if choix ==3 :
    print("Choix de boisson : Café long non sucré\n")
if choix ==4 :
    print("Choix de boisson : Café long sucré\n")
```

```
1 : Café court non sucré
2 : Café court sucré
3 : Café long non sucré
4 : Café long sucré
```

Choix de boisson : Café long non sucré

### 3. Sortie PDF

# Documenter un projet avec Gitlab

Interface web open source basée sur le protocole Git

→ Versionnement des fichiers texte (dépôt local ou distant)

→ Très utile pour la gestion des projets informatiques

Depuis 2019, instance Huma-Num : [gitlab.huma-num.fr](https://gitlab.huma-num.fr)

→ Accessible à partir du portail [Humanid](#)

→ Stockage des données en France (à l'INP23)

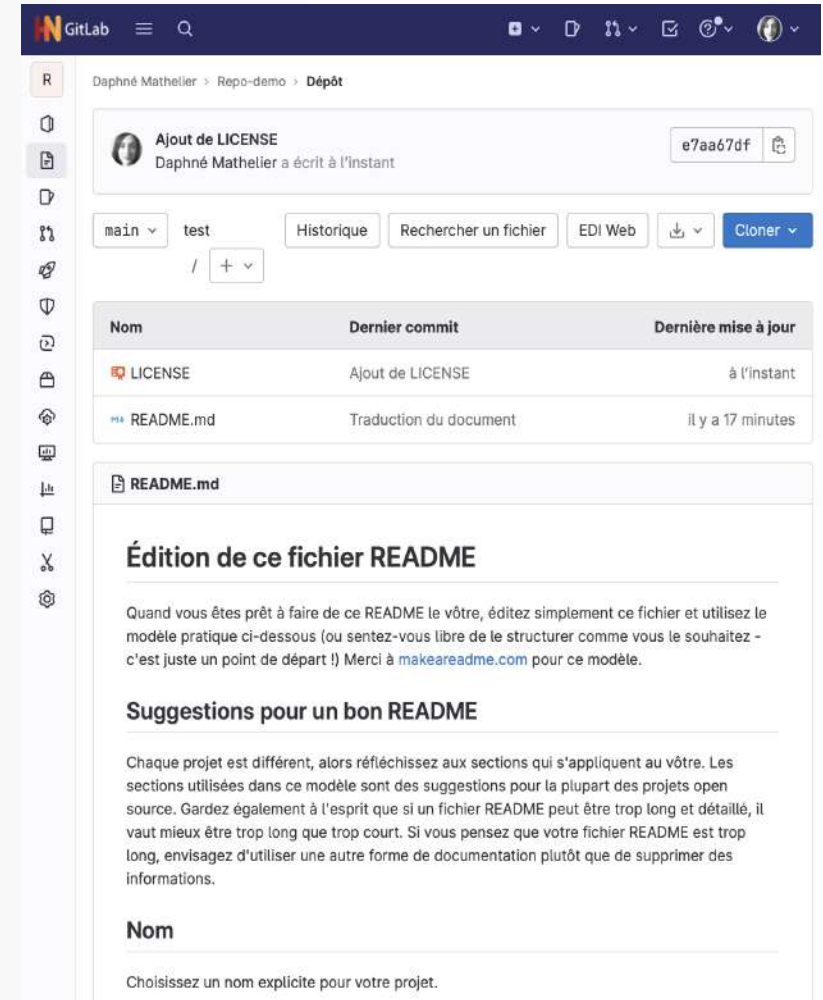
Démarche d'ouverture des projets :

→ Diffusion des dépôts publics dans la plateforme [CodeGouv](#)

→ Archivage dans l'archive des logicielles [Software Heritage](#)

Trois niveaux de visibilité : public, public avec authentification et privé (accès uniquement aux membres du projet)

Regroupement possible de plusieurs projets au sein d'un groupe



GitLab

+ 📄 🔗 ✉ ? 👤

R Repo-demo

🔍 Rechercher sur GitLab

- 📁 Informations du projet
- 📁 Dépôt
- 📁 Fichiers
- 📁 Commits
- 📁 Branches
- 📁 Étiquettes
- 📁 Contributeurs
- 📁 Graphe
- 📁 Comparer
- 📁 Tickets 0
- 🔗 Demandes de fusion 0
- 🔗 Intégration et livraison con...
- 🛡 Sécurité et Conformité
- 📦 Déploiements
- 📦 Paquets & registres
- 🏗 Infrastructure
- 📊 Supervision
- 📊 Analyse
- 📖 Wiki
- ✂ Extraits de code
- ⚙ Paramètres

Daphné Mathelier > Repo-demo > Dépôt

Ajout de LICENSE

Daphné Mathelier a écrit à l'instant

e7aa67df
📄

main ▾ test / + ▾
Historique
Rechercher un fichier
EDI Web
📄 ▾
Cloner ▾

Nom	Dernier commit	Dernière mise à jour
📄 LICENSE	Ajout de LICENSE	à l'instant
📄 README.md	Traduction du document	il y a 17 minutes

📄 README.md

## Édition de ce fichier README

Quand vous êtes prêt à faire de ce README le vôtre, éditez simplement ce fichier et utilisez le modèle pratique ci-dessous (ou sentez-vous libre de le structurer comme vous le souhaitez - c'est juste un point de départ !) Merci à [makeareadme.com](https://makeareadme.com) pour ce modèle.

### Suggestions pour un bon README

Chaque projet est différent, alors réfléchissez aux sections qui s'appliquent au vôtre. Les sections utilisées dans ce modèle sont des suggestions pour la plupart des projets open source. Gardez également à l'esprit que si un fichier README peut être trop long et détaillé, il vaut mieux être trop long que trop court. Si vous pensez que votre fichier README est trop long, envisagez d'utiliser une autre forme de documentation plutôt que de supprimer des informations.

### Nom

Choisissez un nom explicite pour votre projet.

- Liste
- Tableaux
- Service d'assistance
- Jalons

## Créer des fichiers de documentation

### Fichier « Lisez-moi » (*readme*) :

- Généralement à la racine du projet
- Informations générales sur le projet, notamment les objectifs et la méthodologie
- Explications sur l'organisation du répertoire (*repository*)

### Fichier « Licence » : informations sur la licence utilisée

### Tickets (*issues*) :

- Permettent de faire remonter des bugs, des questions, des retours d'expérience, etc.
- S'affichent sous forme de liste, tableau ou jalons
- Outil pour la collaboration et la gestion de projet

### Répertoire de documentation (*wiki*) :

Documentation interne au sein d'un groupe ou d'un projet

The screenshot shows the GitLab web interface for a repository named 'Repo-demo'. The 'README.md' file is open in the editor. The left sidebar shows the repository structure with 'LICENSE' and 'README.md' files. The main editor area has a 'Créer un commit...' button and a '0 fichier modifié' status. The right pane shows a preview of the README content, which includes sections for 'Édition de ce fichier README', 'Suggestions pour un bon README', 'Nom', 'Description', 'Badges', 'Visuels', 'Installation', and 'Utilisation'.

```

1 # Édition de ce fichier README
2
3 Quand vous êtes prêt à faire de ce README le vôtre, éditez simplement
4 ce fichier et utilisez le modèle pratique ci-dessous (ou sentez-vous
5 libre de le structurer comme vous le souhaitez - c'est juste un point
6 de départ !) Merci à [makeareadme.com](https://www.makeareadme.com/)
7 pour ce modèle.
8
9 ## Suggestions pour un bon README
10 Chaque projet est différent, alors réfléchissez aux sections qui
11 s'appliquent au vôtre. Les sections utilisées dans ce modèle sont des
12 suggestions pour la plupart des projets open source. Gardez également
13 à l'esprit que si un fichier README peut être trop long et détaillé,
14 il vaut mieux être trop long que trop court. Si vous pensez que votre
15 fichier README est trop long, envisagez d'utiliser une autre forme de
16 documentation plutôt que de supprimer des informations.
17
18 ## Nom
19 Choisissez un nom explicite pour votre projet.
20
21 ## Description
22 Faites savoir aux gens ce que votre projet peut faire spécifiquement.
23 Fournissez le contexte et ajoutez un lien vers toute référence que
24 les visiteurs pourraient ne pas connaître. Vous pouvez également
25 ajouter ici une liste de caractéristiques ou une sous-section sur le
26 contexte. S'il existe des alternatives à votre projet, c'est un bon
27 endroit pour énumérer les facteurs de différenciation.
28
29 ## Badges
30 Sur certains LISEZMOI, vous pouvez voir de petites images qui
31 transmettent des métadonnées, comme le fait que tous les tests du
32 projet sont réussis ou non. Vous pouvez utiliser Shields pour en
33 ajouter à votre README. De nombreux services ont également des
34 instructions pour ajouter un badge.
35
36 ## Visuels
37 Selon ce que vous réalisez, il peut être judicieux d'inclure des
38 captures d'écran ou même une vidéo (vous verrez souvent des GIF
39 plutôt que de véritables vidéos). Des outils comme ttygif peuvent
40 vous aider, mais consultez Ascinema pour une méthode plus
41 sophistiquée.
42
43 ## Installation
44 Dans un écosystème particulier, il peut y avoir une façon commune
45 d'installer les choses, comme utiliser Yarn, NuGet ou Homebrew.
46 Cependant, tenez compte du fait que la personne qui lit votre fichier
47 README est un novice et qu'elle a besoin de conseils. L'énumération
48 d'étapes spécifiques permet de lever toute ambiguïté et d'amener les
49 gens à utiliser votre projet aussi rapidement que possible. S'il ne
50 fonctionne que dans un contexte spécifique, comme une version de
51 langage de programmation ou un système d'exploitation particulier, ou
52 s'il a des dépendances qui doivent être installées manuellement,
53 ajoutez également une sous-section Exigences.
54
55 ## Utilisation
  
```

## Créer un site de documentation

**GitLab Pages** : fonctionnalité qui permet de générer et de publier, gratuitement et en continu, des sites web statiques associés à des répertoires GitLab

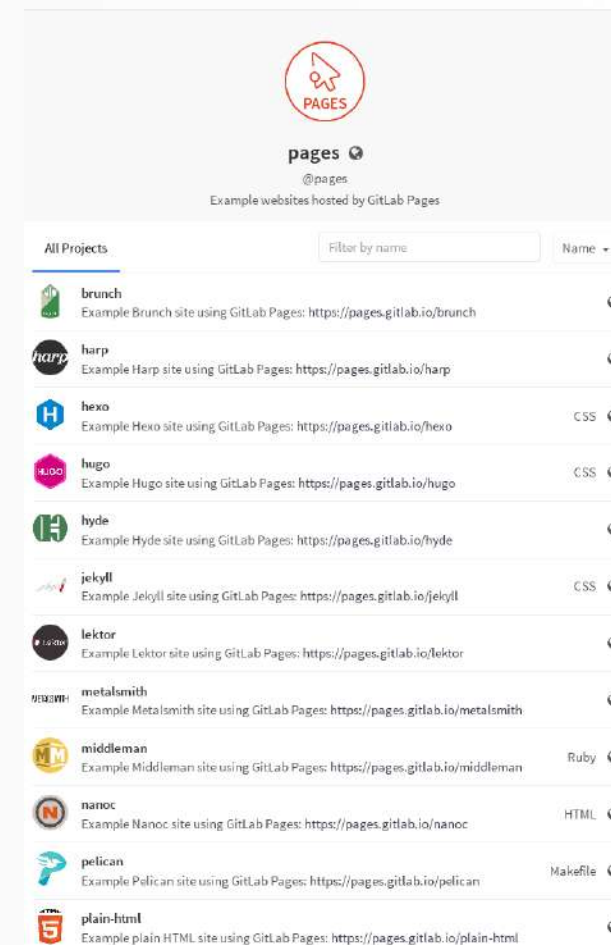
Sous-domaine : `gitpages.huma-num.fr`

### Fonctionnement

- Alimentation du site avec des fichiers textes `md` (ou `html`)
- Gestion des sauvegardes/versions avec `GitLab` (*impératif*)
- Publication avec des générateurs de site statique (*facultatif*)

### Avantages

- Gain de temps et de ressources
- Nombre de sites hébergés non limité
- Adapté aux sites minimalistes ou principalement textuels



## Marche à suivre

Dans GitLab, **créer un nouveau projet**, à partir d'un modèle, par ex. :

Pages/GitBook | Pages/Hugo | Pages/Jekyll | Pages/Plain HTML

### Compléter les informations :

- Le nom du projet
- Le diminutif (*slug*) dans l'adresse URL
- La description du projet
- Le niveau de visibilité public

En fonction de la technologie choisie, **quelques réglages**, par ex. :

- HTML : lancer un premier pipeline pour initialiser le projet
- HUGO : configurer le fichier `config.toml`  
→ Pas de panique, beaucoup de ressources en ligne !

**Répertoire** : `https://gitlab.huma-num.fr/[nom-utilisateur]/[nom-projet]`

**Site public** : `https://[nom-utilisateur].gitpages.huma-num.fr/[nom-projet]`

GitLab

Créer à partir d'un modèle

Créer un projet pré-rempli avec les fichiers nécessaires pour vous permettre de démarrer rapidement.

Nouveau projet > Créer à partir d'un modèle

Apprenez comment [contribuer aux modèles intégrés](#)

**Modèle**

Pages/HTML brut Changer de modèle

**Nom du projet**

Mon projet

**URL du projet** Identifiant « slug » du projet

https://gitlab.hu... dmathelier mon-projet

Vous souhaitez organiser plusieurs projets interdépendants sous un même espace de noms ? [Créer un groupe.](#)

**Description du projet (facultative)**

Mon projet de site avec GitLab Pages basé sur des pages web simple (HTML ou MD)

**Niveau de visibilité** ?

Privé  
L'accès au projet doit être explicitement accordé à chaque utilisateur. Si ce projet fait partie d'un groupe, l'accès est accordé aux membres du groupe.

Interne  
Le projet est accessible à tout utilisateur connecté, à l'exception des utilisateurs externes.

Public  
Votre projet est accessible sans aucune authentification.

Créer le projet Annuler

## Exemples

### Pour s'exercer

- Cours : <https://debugue.ecrituresnumeriques.ca/seance-09-fabriquer-des-sites-web/>
- Répertoire : <https://gitlab.huma-num.fr/ecrinum/demarreur>
- Rendu : <https://ecrinum.gitpages.huma-num.fr/demarreur/>
- Crédits : Chaire du Canada sur les écritures numériques (CCEN)

### Exemple de site de documentation

- Répertoire : <https://gitlab.huma-num.fr/ecrinum/stylo/xml-documentation>
- Rendu : <https://ecrinum.gitpages.huma-num.fr/stylo/xml-documentation/>
- Crédits : Daphné Mathelier & CCEN

### Exemple de présentation

- Répertoire : <https://gitlab.com/formations-kgaut/poc-presentation>
- Rendu : <https://formations-kgaut.gitlab.io/poc-presentation>
- Crédits : Kevin Gautreau

Documentation XML : de Stylo à Érudit	Documentation XML : de Stylo à Érudit
Rechercher	Bienvenue sur la documentation des exports XML de Stylo à Érudit !
Affiliation auteur-e	Cette documentation a pour objectif de lister et de corriger les problèmes rencontrés lors de la validation des XML à destination de la plateforme Érudit.
Alinea	Trois revues utilisent actuellement Stylo pour produire leurs numéros et générer les fichiers XML demandés par Érudit :
Bibliographie	<ul style="list-style-type: none"> <li>• <a href="#">Nouvelles vues</a></li> <li>• <a href="#">Romanticism on the Net</a></li> <li>• <a href="#">Sens public</a></li> </ul>
Citation	Les éléments qui peuvent avoir besoin d'être vérifiés ou corrigés ont été listés dans le menu latéral et <a href="#">ici</a> .
Code	Les problèmes de validation rencontrés peuvent concerner les trois revues ou une seule d'entre elles, en fonction des spécificités de chacune. Chaque page de résolution indique la ou les revue(s) concernée(s) et renvoie au passage adéquat de la <a href="#">documentation d'Érudit</a> .
Dédicace	Vous trouverez également, au sein de cette documentation, des rappels sur le <a href="#">dossier zippé</a> à transmettre à Érudit et sur les <a href="#">scripts</a> à installer et utiliser pour effectuer les exports et la validation des XML.
Descripteur	
Droits d'auteur	
Épigraphe	
Figure	
Genre des directeurs trices	
Identifiant du numéro	
Identifiant propriétaire	
Langue	
Liste	
Mot-clé	
Notice biographique	
Numéro du dossier	
Numéro de figure	
Numéro de revue	
Pagination	
Période de publication	
Tableau	
Type d'article	



# Ressources

## Apprendre la syntaxe

- CommonMark, [Tutorial](#) et sa [traduction française](#) (Arthur Perret)
- GitHub Docs, [Basic writing and formatting syntax](#)
- Markdown Guide, [Basic Syntax](#) et [Cheat Sheet](#)

## Articles et billets

- Dehut J. (2018), [En finir avec Word !](#)
- Fauchier A. (2018), [Markdown comme condition d'une norme de l'écriture numérique](#)

- Pochet B. (2022), [Markdown & vous](#)
- Ruiz E. (2020), [Apprendre à écrire](#)

## Formations

- Bonjour J.-D. (2014), [Élaboration et conversion de documents avec Markdown et Pandoc](#)
- Casajus N. (2016), [Langage Pandoc's Markdown](#)
- Rabaud J. (2021), [Prendre des notes et rédiger avec Markdown](#)
- Simpkin S. (2020), [Débuter avec Markdown](#)
- Tenen D., Wythoff G. (2020), [Rédaction durable avec Pandoc et Markdown](#)
- Université de Montréal (2020), [Initiation à la rédaction Markdown](#)

## Outils présentés

GitLab  
(instance Huma-Num)

[Documentation](#)  
[Démo](#)

GitLab Pages  
[Documentation](#)  
[Démo](#)

GitLab Pages + Hugo  
[Documentation](#)  
[Démo](#)

HedgeDoc  
[Documentation](#)  
[Démo](#)

HedgeDoc + reveal.js  
[Documentation](#)  
[Démo](#)

Jupyter Notebook  
[Documentation](#)  
[Démo](#)

Obsidian  
[Documentation](#)  
[Démo](#)

Obsidian + reveal.js  
[Documentation](#)  
[Démo](#)

Obsidian + Zotero  
[Documentation](#)  
[Démo](#)

Rstudio + reveal.js  
[Documentation](#)  
[Démo](#)

Rstudio + Rmarkdown  
[Documentation](#)  
[Démo](#)

ShareDocs  
[Documentation](#)  
[Démo](#)

Stylo  
[Documentation](#)  
[Démo](#)

VSCodium  
[Documentation](#)  
[Démo](#)

VSCodium + Marp  
[Documentation](#)  
[Démo](#)

VSCodium + reveal.js  
[Documentation](#)  
[Démo](#)

Zettlr  
[Documentation](#)  
[Démo](#)

Zettlr + reveal.js  
[Documentation](#)  
[Démo](#)

Zotero + Better BibTex  
[Documentation](#)  
[Démo](#)