

INTRODUCTION

On va voir **pourquoi documenter est important, ce qu'est Markdown, et comment il simplifie l'écriture d'une documentation claire, lisible, et compatible avec des plateformes comme GitHub.**

C'est un outil simple mais super puissant, et vous verrez que vous pouvez l'utiliser même sans être expert en programmation.

Parfois, lorsqu'on travaille sur un projet logiciel, ce qui est clair pour nous le jour où on code, devient flou deux semaines plus tard... Et c'est là qu'intervient la documentation.

Markdown, c'est un petit langage simple, qu'on utilise partout, même sans s'en rendre compte.

Historique

a. Origines

Markdown a été inventé en **2004** par **John Gruber**, un blogueur et développeur américain.

Il voulait un moyen d'écrire sur internet sans se prendre la tête avec des balises HTML compliquées.

Il a été aidé par **Aaron Swartz**, un prodige de l'informatique connu pour son rôle dans la création de Reddit et RSS.

Ensemble, ils ont créé Markdown, un langage **simple, rapide à lire**, mais capable de produire du **HTML propre** en sortie.

L'idée de base était simple : "Si tu veux un titre, commence par #. Si tu veux du gras, mets **.

b. Évolution au fil des années

Au départ, Markdown était juste un petit outil pour écrire des articles de blog.

Mais rapidement, il a été **adopté dans le monde open source**, notamment sur **GitHub**, où tous les projets utilisent README.md.

Il y a eu ensuite plusieurs **variantes** du langage, car Markdown n'était pas strictement standardisé.

Pour résoudre ça, une version plus stable a été proposée : **CommonMark**, adoptée par beaucoup d'éditeurs.

GitHub a même créé sa propre version, appelée **GitHub Flavored Markdown (GFM)**, qui ajoute des fonctionnalités comme les tableaux, les checklists, etc.

Aujourd'hui, Markdown est utilisé dans :

- Les blogs,
- Les commentaires de code,
- Les éditeurs comme **VS Code, Obsidian, Notion**,

- Les générateurs de documentation comme **MkDocs**, **Docusaurus**, etc.

c. Impact sur la communication

Markdown a eu un **énorme impact sur la façon dont on communique dans les projets informatiques**.

Grâce à lui, on peut rédiger :

- de la documentation lisible même sans logiciel spécial,
- des rapports clairs en version texte,
- des fichiers collaboratifs dans GitHub, Notion, ou Obsidian.

Il est **rapide à apprendre**, utilisable par **tout le monde**, même ceux qui ne sont pas développeurs.

Il facilite donc la **collaboration entre les équipes techniques et non techniques** (rédacteurs, designers, gestionnaires de projet...).

Bref, Markdown est devenu une **langue commune** entre humains et machines pour **structurer et partager l'information simplement**.

Utilité

a- Rôle dans le développement logiciel

Dans le développement logiciel, Markdown est partout.

Il est utilisé pour rédiger les **README.md**, ces fichiers qu'on trouve à la racine de chaque projet sur GitHub.

On y explique : comment installer le projet, quelles sont ses fonctionnalités, comment y contribuer, etc.

On utilise aussi Markdown pour :

- Écrire des **journaux de modifications** (*changelogs*),
- Documenter les **APIs**,
- Ajouter des **commentaires techniques** dans certains outils.

Bref, c'est devenu un outil **standard dans les projets de développement**, car il permet de **garder une trace écrite claire et versionnée** avec le code.

b- Importance dans la documentation

Markdown est particulièrement utile pour écrire de la **documentation technique**, car :

- Il permet d'écrire un texte **structuré, clair, et hiérarchisé**.
- Le même document peut être **lu brut dans un éditeur de texte**, ou **converti en HTML** automatiquement.
- On peut facilement y insérer des **titres, des listes, du code, des liens, des images...**

Et comme Markdown est souvent utilisé avec **Git**, la documentation peut évoluer **en même temps que le code**.

Résultat : une documentation **toujours à jour**, intégrée au projet.

c- Facilitation de la collaboration

Markdown joue un grand rôle dans la **collaboration en équipe**.

Comme il est **facile à apprendre, tout le monde peut l'utiliser** :

- Développeurs pour le code et la doc
- Rédacteurs pour les notes ou guides
- Chefs de projet pour suivre les instructions techniques

Sur des plateformes comme **GitHub, GitLab, Notion**, Markdown devient un **langage commun** entre les membres d'une équipe.

Résultat : tout le monde peut contribuer à un même document, sans se battre avec la mise en page ou des outils compliqués.

Transition pour la suite

On a donc vu **d'où vient Markdown** et **à quoi il sert**, maintenant, intéressons-nous à la **syntaxe de base**, c'est-à-dire : **comment on l'utilise concrètement pour écrire**.

Utilité

a- Architecture du langage Markdown

Markdown, c'est un **langage de balisage**, un peu comme HTML, sauf qu'il est **beaucoup plus simple et lisible**.

Il n'est **pas exécuté** comme un langage de programmation. Il est **interprété** :

- Tu écris du texte avec une **syntaxe claire** (# pour les titres, ** pour le gras, etc.)
- Ensuite, un outil ou un moteur de rendu (comme GitHub, VS Code, Obsidian, Docusaurus...) le **transforme en HTML**

Le rôle de Markdown, c'est de **structurer ton texte**, pas de l'exécuter. Il agit donc comme une **couche entre l'auteur et le format final**.

b- Fonctionnement général de Markdown

Quand tu écris du Markdown, ce n'est pas ce texte que voit l'utilisateur final.

Il passe par un **moteur de conversion** qui le transforme en **HTML**.

Quelques moteurs populaires :

- **Marked.js** (JavaScript)
- **Pandoc** (convertit en PDF, Word, HTML...)
- **CommonMark Parser** (standard moderne)

Ces moteurs lisent ton texte .md, analysent la structure (#, *, [](), etc.), et produisent du **code HTML** derrière.

Exemple :

Bonjour== <h1>Bonjour</h1>