

MarkItDown sous Windows

Guide complet d'installation et d'utilisation - avec « Convertir en Markdown V11 »

Pour Windows 10 et Windows 11

Version du tutoriel : août 2026

Ce guide couvre deux parcours :

- PC équipé d'un Python Windows classique ou d'un Python portable ;
- PC ne disposant pas encore de Python.
- Il explique ensuite l'installation et toutes les fonctions de la V11 : lots, formats mélangés, glisser-déposer, dossiers et sous-dossiers, ODT, WPD, suivi de progression, annulation, rapports, fenêtre finale des résultats et regroupement des fichiers Markdown dans un dossier unique.

Document pratique destiné aux utilisateurs débutants

Sommaire

- Partie I - Comprendre MarkItDown et identifier la configuration du PC
- Partie II - Installer MarkItDown selon la configuration du PC
- Partie III - Vérifier l'installation et comprendre l'avertissement FFmpeg
- Partie IV - Installer « Convertir en Markdown V11 »
- Partie V - Utiliser toutes les fonctions de la V11
- Partie VI - Formats particuliers : ODT, WPD et PDF numérisés
- Partie VII - Dépannage et procédures résumées
- Annexe - Ressources officielles et checklist finale

Conseil

Pour une première installation, suivre le guide dans l'ordre. Une fois la V11 installée, l'usage quotidien ne nécessite normalement plus PowerShell.

Partie I - Comprendre MarkItDown et identifier la configuration du PC

1. Objet de ce guide

Ce guide explique, pas à pas, comment installer MarkItDown sous Windows puis comment ajouter l'outil complémentaire « Convertir en Markdown V11 », conçu pour utiliser MarkItDown sans saisir de commandes au quotidien.

Deux situations sont distinguées :

- le PC possède déjà Python, soit sous la forme d'une installation Windows classique, soit sous la forme d'un Python portable ;
- le PC ne possède pas encore Python : il faut alors commencer par l'installer.

2. Qu'est-ce que MarkItDown ?

MarkItDown est un outil Python développé par Microsoft qui transforme différents types de documents en fichiers Markdown (.md). Le format Markdown est un texte structuré, léger et particulièrement utile pour l'indexation, la recherche plein texte, l'analyse documentaire et l'exploitation par des outils d'intelligence artificielle.

MarkItDown peut notamment traiter des documents Word, PDF, PowerPoint, Excel, HTML et plusieurs formats textuels. La V11 ajoute en complément un traitement local du format ODT et une voie de conversion pour les fichiers WordPerfect WPD lorsque LibreOffice est disponible.

3. Version de Python recommandée

MarkItDown exige Python 3.10 ou supérieur. Pour une nouvelle installation destinée à MarkItDown, ce guide recommande Python 3.13 : il s'agit d'un choix prudent et largement compatible avec l'installation complète « markitdown[all] ».

```
Python 3.13
```

4. Première étape : déterminer la situation du PC

Ouvrir Windows PowerShell : menu Démarrer, saisir « PowerShell », puis ouvrir Windows PowerShell. Les droits administrateur ne sont normalement pas nécessaires pour les vérifications ci-dessous.

4.1. Vérifier si Python est déjà installé normalement

```
python --version
```

Si une réponse comparable à « Python 3.13.x » apparaît, Python est déjà installé. On peut également essayer :

```
py --version
```

Si l'une des deux commandes fonctionne et indique Python 3.10 ou supérieur, poursuivre avec la Partie II, parcours A.

4.2. Si les commandes python et py ne sont pas reconnues

Cela ne signifie pas nécessairement que Python est absent. Il peut exister sous forme portable.
Rechercher un dossier tel que :

```
C:\Users\NomUtilisateur\Downloads\python-portable
```

Dans ce dossier, rechercher « python.exe ». S'il existe, poursuivre avec la Partie II, parcours B.

4.3. Si aucun Python n'est présent

Poursuivre avec la Partie II, parcours C : installation d'un Python Windows classique.

Partie II - Installer MarkItDown selon la configuration du PC

5. Parcours A - Le PC possède déjà un Python Windows classique

5.1. Vérifier la version de Python

```
python --version
```

La version doit être Python 3.10 ou supérieure.

5.2. Vérifier pip

pip est le gestionnaire utilisé pour installer les modules Python.

```
python -m pip --version
```

Une ligne indiquant la version de pip et son emplacement doit apparaître.

5.3. Mettre à jour les outils d'installation

```
python -m pip install --upgrade pip
```

```
python -m pip install --upgrade setuptools wheel
```

5.4. Installer MarkItDown

```
python -m pip install "markitdown[all]"
```

Conserver les guillemets. L'option [all] installe les dépendances optionnelles permettant de prendre en charge le plus grand nombre de formats. Pendant l'installation, de nombreuses lignes « Collecting », « Downloading » et « Installing collected packages » peuvent apparaître : c'est normal.

5.5. Vérifier que MarkItDown est importable

```
python -W ignore -c "import markitdown; print('MarkItDown est correctement installé')"
```

Le résultat attendu est :

```
MarkItDown est correctement installé
```

6. Parcours B - Le PC possède un Python portable

Supposons que Python portable se trouve dans le dossier suivant ; adapter le chemin si nécessaire :

```
C:\Users\NomUtilisateur\Downloads\python-portable
```

6.1. Définir le chemin dans PowerShell

```
$PYDIR = "C:\Users\NomUtilisateur\Downloads\python-portable"
$PY = "$PYDIR\python.exe"
```

6.2. Vérifier l'existence de python.exe

```
Test-Path $PY
```

Le résultat attendu est « True ».

6.3. Vérifier la version

```
& $PY --version
```

6.4. Vérifier pip

```
& $PY -m pip --version
```

Si une version de pip s'affiche, poursuivre directement au point 6.6.

6.5. Cas particulier : « No module named pip »

Certaines éditions portables ou « embeddable » de Python ne contiennent pas pip. Il faut alors l'activer et l'installer.

Rechercher d'abord un fichier de type python313._pth :

```
Get-ChildItem "$PYDIR\python*._pth"
```

Si un fichier apparaît, le mémoriser et en faire une copie de sécurité :

```
$PTH = Get-ChildItem "$PYDIR\python*._pth" | Select-Object -First 1
Copy-Item $PTH.FullName "$($PTH.FullName).backup"
```

Afficher son contenu :

```
Get-Content $PTH.FullName
```

Si la ligne « #import site » est présente, l'activer :

```
(Get-Content $PTH.FullName) -replace '^s*#\s*import site\s*$', 'import site' | Set-Content
$PTH.FullName -Encoding ASCII
```

Télécharger ensuite l'installateur officiel de pip :

```
Invoke-WebRequest -Uri "https://bootstrap.pypa.io/get-pip.py" -OutFile "$PYDIR\get-pip.py"
```

Puis lancer :

```
& $PY "$PYDIR\get-pip.py"
```

Vérifier enfin :

```
& $PY -m pip --version
```

6.6. Mettre à jour les composants Python

```
& $PY -m pip install --upgrade pip
& $PY -m pip install --upgrade setuptools wheel
```

6.7. Installer MarkItDown dans le Python portable

```
& $PY -m pip install "markdown[all]"
```

6.8. Vérifier l'installation

```
& $PY -W ignore -c "import markdown; print('MarkItDown est correctement installé')"
```

Point important

Avec un Python portable, il faut installer MarkItDown dans CE Python portable. Installer MarkItDown dans un autre Python présent sur le PC ne rend pas automatiquement le module disponible dans la version portable.

7. Parcours C - Aucun Python n'est installé sur le PC

La solution la plus simple consiste à installer Python 3.13 sous Windows.

7.1. Vérifier WinGet

```
winget --version
```

7.2. Installer Python 3.13 avec WinGet

```
winget install --exact --id Python.Python.3.13 --source winget
```

Attendre la fin de l'installation, puis fermer complètement PowerShell et ouvrir une nouvelle fenêtre pour que Windows actualise les chemins des programmes.

7.3. Vérifier Python

```
python --version
```

7.4. Installer MarkItDown

```
python -m pip install --upgrade pip
```

```
python -m pip install --upgrade setuptools wheel  
python -m pip install "markdown[all]"
```

7.5. Vérifier l'installation

```
python -W ignore -c "import markdown; print('MarkItDown est correctement installé')"
```

Si WinGet est absent

Installer Python 3.13 depuis la source officielle Python ou le Microsoft Store, puis fermer et rouvrir PowerShell avant de reprendre les commandes.

Partie III - Vérifier l'installation et comprendre l'avertissement FFmpeg

8. Premier test manuel de MarkItDown

Avant d'installer la V11, il est utile de vérifier une conversion simple. Avec un Python Windows classique, par exemple :

```
markitdown "C:\Users\NomUtilisateur\Documents\essai.pdf" -o "C:\Users\NomUtilisateur\Documents\essai.md"
```

Avec Python portable :

```
& "$PYDIR\Scripts\markitdown.exe" "C:\Users\NomUtilisateur\Documents\essai.pdf" -o "C:\Users\NomUtilisateur\Documents\essai.md"
```

Le fichier essai.md doit apparaître dans le même dossier que le PDF.

9. Avertissement FFmpeg

Lors de certaines opérations, le message suivant peut apparaître :

```
RuntimeWarning: Couldn't find ffmpeg or avconv - defaulting to ffmpeg, but may not work
```

Cet avertissement vient notamment du composant pydub. Il ne signifie pas que MarkItDown est défectueux. Pour l'usage documentaire courant (PDF, DOCX, ODT, PPTX, XLSX, etc.), FFmpeg n'est généralement pas nécessaire ; il concerne surtout certaines fonctions audio.

Dans la V11

L'installateur et le backend issus de la V9 stable ignorent cet avertissement lorsqu'il est sans incidence sur la conversion documentaire.

Partie IV - Installer « Convertir en Markdown V11 »

10. À quoi sert la V11 ?

La V11 est une interface locale qui évite de saisir des commandes PowerShell pour chaque conversion. Elle conserve le moteur Python stable validé avec les versions précédentes et ajoute la sélection graphique, les lots, le glisser-déposer, le traitement des dossiers ainsi qu'une fenêtre finale permettant de retrouver, ouvrir et regrouper tous les fichiers Markdown produits.

11. Utiliser l'archive V11

Utiliser l'archive « MarkItDown_V11_Resultats_Regroupees.zip ». L'installateur recherche automatiquement un Python Windows classique ou un Python portable dans les emplacements usuels, puis vérifie que MarkItDown y est installé.

Décompresser entièrement l'archive avant de l'exécuter :

1. Clic droit sur le fichier ZIP.
2. Choisir « Extraire tout... ».
3. Ouvrir le dossier extrait.
4. Double-cliquer sur INSTALLER.cmd.

```
INSTALLER.cmd
```

12. Si l'installateur trouve Python mais pas MarkItDown

Il faut installer MarkItDown dans le Python concerné, puis relancer INSTALLER.cmd. Par exemple, avec un Python classique :

```
python -m pip install "markitdown[all]"
```

Ou avec un Python portable déjà défini dans \$PY :

```
& $PY -m pip install "markitdown[all]"
```

13. Ce que la V11 installe

La V11 crée principalement le dossier suivant dans le profil utilisateur :

```
C:\Users\NomUtilisateur\AppData\Local\MarkItDown-Un-Clic
```

Elle y place le contrôleur PowerShell, le backend Python et la configuration du Python utilisé. Elle crée également :

- une icône « Convertir en Markdown » sur le Bureau ;
- une entrée « Convertir en Markdown » dans le menu Windows « Envoyer vers ».

Aucun service de conversion en ligne n'est nécessaire au fonctionnement normal du dispositif.

Partie V - Utiliser toutes les fonctions de la V11

14. Conversion d'un fichier unique

Double-cliquer sur l'icône « Convertir en Markdown », ajouter ou déposer le document, puis cliquer sur « Convertir en Markdown ». Un fichier decision.pdf produira decision.md dans le même dossier.

15. Conversion en lots

La V11 accepte plusieurs fichiers en une seule opération. Il est possible de sélectionner quelques documents ou plusieurs dizaines de fichiers avant de lancer une conversion unique.

16. Formats différents dans le même lot

Les fichiers n'ont pas besoin d'avoir le même format. Un même lot peut mélanger PDF, DOCX, ODT, WPD, PPTX, XLSX, HTML, TXT, CSV, JSON, XML, EPUB, MSG ou ZIP. Le backend choisit automatiquement le traitement applicable à chaque fichier.

17. Glisser-déposer dans la fenêtre V11

Après un double-clic sur l'icône du Bureau, une zone de glisser-déposer apparaît. On peut y déposer :

- un fichier ;
- plusieurs fichiers ;
- un dossier entier ;
- plusieurs dossiers ;
- un mélange de fichiers et de dossiers.

18. Ajouter des fichiers avec le bouton

Le bouton « Ajouter des fichiers... » ouvre la fenêtre Windows classique. Maintenir Ctrl pour sélectionner des fichiers séparés ou Maj pour sélectionner une plage.

19. Convertir un dossier entier

Le bouton « Ajouter un dossier... » permet de sélectionner un dossier. V11 recherche alors automatiquement les fichiers dont les extensions sont prises en charge.

```
Dossier Affaire
├── conclusions.docx
├── expertise.pdf
├── notes.odt
├── calcul.xlsx
└── Jurisprudence
    ├── arret1.pdf
    └── arret2.pdf
```

20. Inclure ou exclure les sous-dossiers

La case « Inclure les sous-dossiers des dossiers sélectionnés » est cochée par défaut. Si elle reste cochée, l'arborescence complète est explorée. Si elle est décochée, seuls les fichiers situés directement dans le dossier choisi sont pris en compte.

21. Glisser directement des fichiers sur l'icône du Bureau

Depuis l'Explorateur Windows, sélectionner un ou plusieurs fichiers puis les déposer directement sur l'icône « Convertir en Markdown ». La conversion démarre automatiquement, sans ouvrir d'abord la fenêtre de sélection.

22. Glisser directement un dossier sur l'icône du Bureau

Un dossier entier peut également être déposé sur l'icône. Dans ce mode, les sous-dossiers sont inclus automatiquement. Cette méthode est particulièrement rapide pour traiter une arborescence complète.

23. Utiliser le menu « Envoyer vers »

Dans l'Explorateur Windows : sélectionner un ou plusieurs fichiers, faire un clic droit, choisir éventuellement « Afficher plus d'options » sous Windows 11, puis « Envoyer vers » > « Convertir en Markdown ».

24. Ne jamais écraser un fichier Markdown existant

Si rapport.md existe déjà, V11 ne l'écrase pas. Le nouveau fichier est nommé rapport_converti_1.md, puis rapport_converti_2.md, etc. Le document source n'est jamais modifié par ce mécanisme.

25. Conservation de l'arborescence

Chaque fichier .md est créé à côté de son original. Ainsi :

```
C:\Dossiers\Affaire X\Expertises\rapport.pdf
→ C:\Dossiers\Affaire X\Expertises\rapport.md
```

La structure des dossiers reste donc naturellement préservée.

26. Où sont enregistrés les fichiers Markdown ?

Par défaut, chaque fichier .md reste enregistré dans le même dossier que son fichier source. Lorsqu'un lot contient des fichiers provenant de plusieurs dossiers, ou lorsqu'un dossier entier avec ses sous-dossiers est converti, les résultats restent donc répartis dans l'arborescence d'origine. Ce choix conserve la correspondance directe entre chaque document source et son fichier Markdown.

Exemple :

```
C:\Dossiers\Affaire\Expertises\rapport.pdf → C:\Dossiers\Affaire\Expertises\rapport.md
```

La V11 ajoute toutefois une fenêtre de résultats afin qu'il ne soit plus nécessaire de parcourir manuellement tous les dossiers pour retrouver les fichiers produits.

27. Fenêtre finale des résultats et regroupement du lot

À la fin de chaque conversion comportant au moins un fichier réussi, une fenêtre « MarkItDown - Résultats du lot » s'ouvre automatiquement. Elle réunit dans une même liste tous les fichiers Markdown créés pendant l'opération, qu'ils proviennent de fichiers multiples, d'un dossier entier ou de plusieurs sous-dossiers.

Pour chaque résultat, la fenêtre indique :

- le nom du fichier Markdown ;
- le dossier dans lequel il est réellement enregistré ;
- le nom du fichier source correspondant.

Les commandes disponibles sont :

- « Ouvrir le fichier » : ouvre directement le fichier .md sélectionné ;
- « Ouvrir son dossier » : ouvre l'Explorateur Windows et sélectionne le .md à son emplacement d'origine ;
- « Voir le rapport » : ouvre le rapport détaillé du dernier lot ;
- « Réunir une copie de tous les .md... » : copie tous les résultats dans un dossier unique choisi par l'utilisateur, sans déplacer ni supprimer les fichiers originaux.

Lors du regroupement, la V11 crée automatiquement un sous-dossier horodaté du type :

MarkItDown_Resultats_2026-08-21_16-45-30

Si plusieurs fichiers portent le même nom, aucun résultat n'est écrasé : la V11 ajoute automatiquement un suffixe, par exemple rapport.md, rapport_2.md, rapport_3.md.

Le dossier regroupé contient également un fichier « INDEX-DES-FICHIERS.txt » qui indique, pour chaque copie Markdown, le chemin du document source correspondant. Cette fonction est particulièrement utile pour réutiliser immédiatement tout un lot dans un outil d'IA, une base documentaire ou un traitement ultérieur.

Important : le regroupement crée des copies. Les fichiers .md d'origine restent à côté de leurs documents sources et l'arborescence initiale n'est pas modifiée.

28. Barre d'activité, progression et moteur utilisé

Pendant une conversion, V11 affiche le numéro du fichier, le nombre total, le nom du fichier en cours, le moteur utilisé, une barre d'activité et le temps écoulé.

Fichier 17 sur 42
Moteur : MarkItDown Python API
Temps écoulé : 5 s

29. Bouton Annuler

Le bouton « Annuler » permet d'interrompre le traitement. Les documents déjà convertis restent disponibles. Un fichier Markdown partiellement créé pendant l'interruption est supprimé lorsque cela est possible.

30. Délai de sécurité par fichier

Pour éviter qu'un document défectueux n'immobilise indéfiniment le lot, un délai maximal d'environ cinq minutes par fichier est appliqué au traitement principal. Si le délai est dépassé, le processus est arrêté et l'échec est enregistré.

300 secondes = 5 minutes

31. Gestion des erreurs dans un lot

L'échec d'un fichier n'annule pas automatiquement le reste du lot. Exemple : 100 fichiers demandés, 97 réussites, 3 échecs. Les 97 résultats valides sont conservés et les erreurs sont consignées dans le rapport.

32. Protection contre les très gros lots

Au-delà de 200 fichiers détectés, V11 demande une confirmation avant de commencer. Cette mesure évite de lancer involontairement la conversion d'une arborescence très volumineuse.

33. Rapport du dernier lot

Le rapport détaillé est enregistré dans le dossier temporaire Windows sous le nom :

```
MarkItDown-dernier-lot.txt
```

Son emplacement est généralement :

```
C:\Users\NomUtilisateur\AppData\Local\Temp\MarkItDown-dernier-lot.txt
```

Le rapport indique les fichiers réussis, les fichiers en échec, le fichier Markdown produit et le moteur utilisé. Depuis la V11, ce rapport est également accessible directement depuis la fenêtre finale des résultats.

34. Diagnostic V11

Un fichier de diagnostic est créé ici :

```
C:\Users\NomUtilisateur\AppData\Local\MarkItDown-Un-Clic\DIAGNOSTIC-V11.txt
```

Il indique notamment le Python sélectionné automatiquement, son dossier, le résultat du test d'import de MarkItDown et les principales fonctions installées, notamment la fenêtre finale de résultats et le regroupement des fichiers Markdown.

Partie VI - Formats particuliers : ODT, WPD et PDF numérisés

35. Conversion ODT

MarkItDown ne prend pas directement en charge ODT dans la configuration utilisée ici. V11 dispose donc d'un convertisseur ODT local intégré au backend Python. Il tente de conserver les paragraphes, titres, listes, tableaux, liens, une partie du gras et de l'italique et les notes simples.

Limite du format Markdown

Markdown ne reproduit pas toute la mise en page d'un document LibreOffice. Les objets complexes, zones de texte, dessins et certains styles avancés peuvent être simplifiés ou perdus.

36. Conversion WordPerfect WPD

MarkItDown ne sait pas lire directement les fichiers WordPerfect .wpd. V11 utilise donc LibreOffice lorsqu'il est installé sur le PC. Le traitement est local :

```
WPD
↓
conversion locale par LibreOffice
↓
ODT temporaire
↓
Markdown
```

Si LibreOffice n'est pas installé, les autres formats continuent de fonctionner ; seuls les fichiers WPD ne peuvent pas être convertis par cette voie.

37. PDF numérisé sans couche texte

Un PDF issu d'un scanner peut être constitué uniquement d'images. Dans ce cas, MarkItDown peut produire peu de texte ou aucun texte parce qu'aucune couche textuelle exploitable n'est présente. Un traitement OCR est alors nécessaire ; il s'agit d'un besoin distinct de la conversion MarkItDown elle-même.

38. Confidentialité et traitement local

Les conversions réalisées par MarkItDown, le convertisseur ODT intégré et LibreOffice sont locales sur le PC. La V11 n'envoie pas les documents à un service de conversion en ligne. Cette architecture est adaptée aux documents professionnels et confidentiels, sous réserve des règles de sécurité propres au poste de travail.

Partie VII - Dépannage et procédures résumées

39. Dépannage rapide

« python n'est pas reconnu »

```
py --version
```

Si cette commande échoue également, rechercher un Python portable ou installer Python 3.13.

« No module named pip »

Le Python portable ne contient pas pip. Suivre la procédure du point 6.5.

« No module named markitdown »

MarkItDown n'est pas installé dans le Python qui vient d'être utilisé. Pour un Python classique :

```
python -m pip install "markitdown[all]"
```

Pour le Python portable défini par \$PY :

```
& $PY -m pip install "markitdown[all]"
```

Message FFmpeg

```
Couldn't find ffmpeg or avconv
```

Ce message peut être ignoré pour les conversions documentaires habituelles ; il ne constitue pas à lui seul un échec.

Le WPD échoue alors que le PDF fonctionne

Vérifier que LibreOffice est installé. PDF et WPD n'utilisent pas exactement le même moteur.

Un fichier échoue mais les autres fonctionnent

Consulter MarkItDown-dernier-lot.txt pour connaître le motif exact et le fichier concerné.

40. Procédure résumée - Python déjà installé

```
python --version
python -m pip install --upgrade pip
python -m pip install "markitdown[all]"
python -W ignore -c "import markitdown; print('MarkItDown est correctement installé')"
```

Puis décompresser la V11 et exécuter INSTALLER.cmd.

41. Procédure résumée - Python portable

```
$PYDIR = "C:\Chemin\python-portable"
$PY = "$PYDIR\python.exe"
& $PY -m pip install --upgrade pip
```

```
& $PY -m pip install "markdown[all]"  
& $PY -W ignore -c "import markdown; print('MarkItDown est correctement installé')"
```

Puis décompresser la V11 et exécuter INSTALLER.cmd.

42. Procédure résumée - Aucun Python

```
winget install --exact --id Python.Python.3.13 --source winget
```

Fermer puis rouvrir PowerShell, puis :

```
python -m pip install --upgrade pip  
python -m pip install "markdown[all]"
```

Enfin, installer V11 avec INSTALLER.cmd.

Annexe - Ressources officielles et checklist finale

Ressources officielles

Projet Microsoft MarkItDown : github.com/microsoft/markitdown

Installation de Python sous Windows : [Documentation Microsoft Windows / Python](#)

Installation de pip pour Python portable : bootstrap.pypa.io/get-pip.py

Checklist finale

- ☐ Python 3.10 ou supérieur est installé.
- ☐ De préférence, Python 3.13 est utilisé pour une nouvelle installation.
- ☐ pip fonctionne.
- ☐ markitdown[all] est installé.
- ☐ Le test import markitdown fonctionne.
- ☐ Le ZIP V11 a été entièrement décompressé.
- ☐ INSTALLER.cmd a été exécuté.
- ☐ L'icône « Convertir en Markdown » apparaît sur le Bureau.
- ☐ Un PDF de test a été converti.
- ☐ Le glisser-déposer fonctionne.
- ☐ Un lot de plusieurs fichiers a été testé.
- ☐ La conversion d'un dossier a été testée.
- ☐ L'option des sous-dossiers a été vérifiée.
- ☐ La fenêtre finale des résultats s'ouvre après une conversion réussie.
- ☐ Le bouton « Ouvrir son dossier » permet de retrouver immédiatement un fichier converti.
- ☐ Le regroupement de tous les .md dans un dossier unique a été testé.
- ☐ Le fichier INDEX-DES-FICHIERS.txt est créé dans le dossier de regroupement.
- ☐ LibreOffice est installé si la conversion de fichiers WPD est nécessaire.

Résultat attendu

Une fois l'installation terminée, l'utilisateur n'a normalement plus besoin de PowerShell pour son usage quotidien : il utilise l'icône « Convertir en Markdown », le glisser-déposer, la sélection de dossiers ou le menu « Envoyer vers ».

Fin du tutoriel