

## PowerShell – `Export-Csv` : exporter un fichier CSV

`Export-Csv` permet d'enregistrer le résultat d'une commande PowerShell dans un **fichier CSV**. Un CSV est un fichier texte où les données sont organisées en colonnes séparées par des virgules ; il s'ouvre facilement avec Excel, LibreOffice ou un éditeur de texte.

6TQ

 niveau

PowerShell n'exporte pas du texte brut : il exporte **des objets**, dont chaque propriété devient une colonne.

## Syntaxe de base

```
1 | <commande> | Export-Csv -Path "fichier.csv"
```

Exemple :

```
1 | Get-Process | Export-Csv -Path "processus.csv"
```

Cela crée un fichier CSV contenant toutes les propriétés de chaque processus.

## Option essentielle : `-NoTypeInfo`

PowerShell ajoute par défaut une première ligne indiquant le type d'objet. Pour un CSV propre (surtout pour Excel), on supprime cette ligne :

```
1 | Get-Process | Export-Csv "processus.csv" -NoTypeInfo
```

# Choisir les colonnes avant export

Utilisez `Select-Object` pour ne garder que les propriétés utiles.

Exemple :

```
1 | Get-ChildItem -File |  
2 |   Select-Object Name, Length, LastWriteTime |  
3 |   Export-Csv "fichiers.csv" -NoTypeInfoation
```

# Exporter en utilisant un autre séparateur (point-virgule)

Pratique pour Excel en français :

```
1 | Export-Csv "data.csv" -Delimiter ";"
```

# Ajouter au fichier au lieu d'écraser

```
1 | Export-Csv "log.csv" -Append -NoTypeInfoation
```

## Résumé

Action	Commande		
Exporter un résultat	`...`	Export-Csv "file.csv"	
Export propre sans entête technique	<code>-NoTypeInfoation</code>		

Action	Commande		
Définir les colonnes	`...`	Select-Object ...	Export-Csv`
Délimiteur personnalisé	<code>-Delimiter ";"</code>		
Ajouter au fichier	<code>-Append</code>		