

EXERCICE N°14

POWERSHELL

Jean-Pierre Duchesneau, Cégep Sainte-Foy, DFC

Automne 2021

Évaluation : formative

Travail de préférence individuel.

Durée : 3 heures

Système d'exploitation : Windows 10 20H2, votre VM Windows Générique

Sites de référence :

- Documentation : <https://docs.microsoft.com/fr-fr/powershell/scripting/overview?view=powershell-7.1>
 - PowerShell Core : <https://github.com/PowerShell/PowerShell>
-

1 Prise en main de PowerShell

Toutes les versions actuelles des systèmes d'exploitation Windows sont livrées avec PowerShell.

1.1 Où trouver PowerShell ?

Le moyen le plus simple de rechercher PowerShell sur Windows 10 est de taper PowerShell dans la barre de recherche, comme le montre la figure 11.

Notez que la figure 11 montre quatre raccourcis différents pour PowerShell. Dans ce livre [PowerShell101], l'ordinateur utilisé à des fins de démonstration exécute la version 64 bits de Windows 10. Vous avez donc une version 64 bits de la console PowerShell et de PowerShell ISE (environnement de script intégré), ainsi qu'une version 32 bits pour chacun, comme l'indique le suffixe (x86) dans les raccourcis. Si vous exécutez une version 32 bits de Windows 10, vous n'aurez que deux raccourcis. Ces éléments n'ont pas le suffixe (x86), mais il s'agit de la version 32 bits. Si vous disposez d'un système d'exploitation 64 bits, je vous recommande d'exécuter la version 64 bits de PowerShell, sauf si vous avez une raison particulière d'exécuter la version 32 bits.

1.2 Comment lancer PowerShell

Si vous lancez PowerShell en cliquant sur le raccourci « Windows PowerShell », comme illustré dans la figure 11. Notez que la barre de titre de la console PowerShell indique « Windows PowerShell », comme dans la figure 12. Certaines commandes s'exécutent bien, toutefois, PowerShell ne peut pas

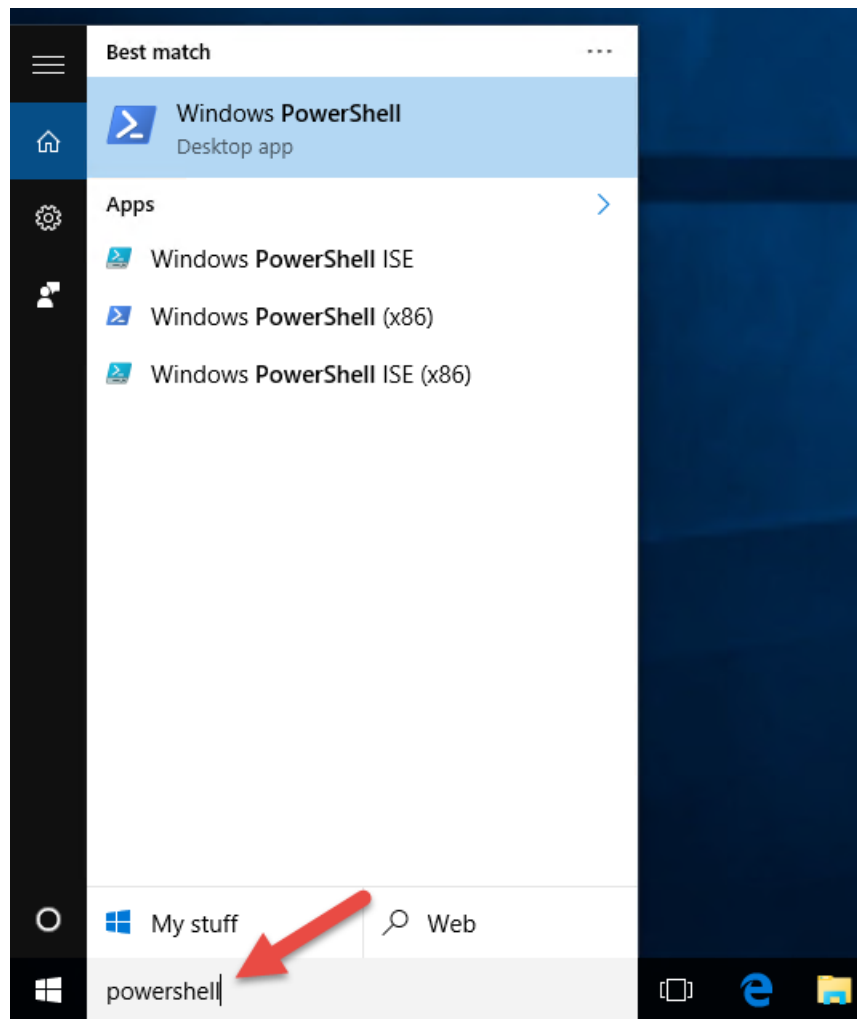


FIGURE 1 – trouver PowerShell

participer au contrôle de compte d'utilisateur (UAC). Cela signifie qu'il ne peut pas demander une élévation des privilèges pour les tâches qui nécessitent l'approbation d'un administrateur. C'est pourquoi

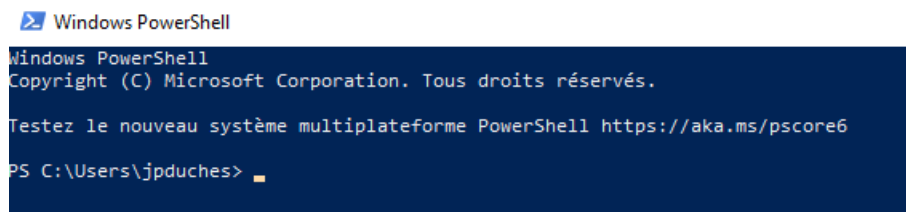


FIGURE 2 – Lancer PowerShell

dans les exercices, je vous invite à lancer PowerShell avec les droits d'administrateur comme à la figure 13

Une fois que vous avez relancé PowerShell en tant qu'administrateur, la barre de titre doit indiquer « Administrateur : Windows PowerShell », comme illustré à la figure 14.

Maintenant que PowerShell est exécuté avec une élévation des privilèges en tant qu'administrateur local, le contrôle de compte d'utilisateur ne posera plus de problème lorsque vous exécuterez une com-

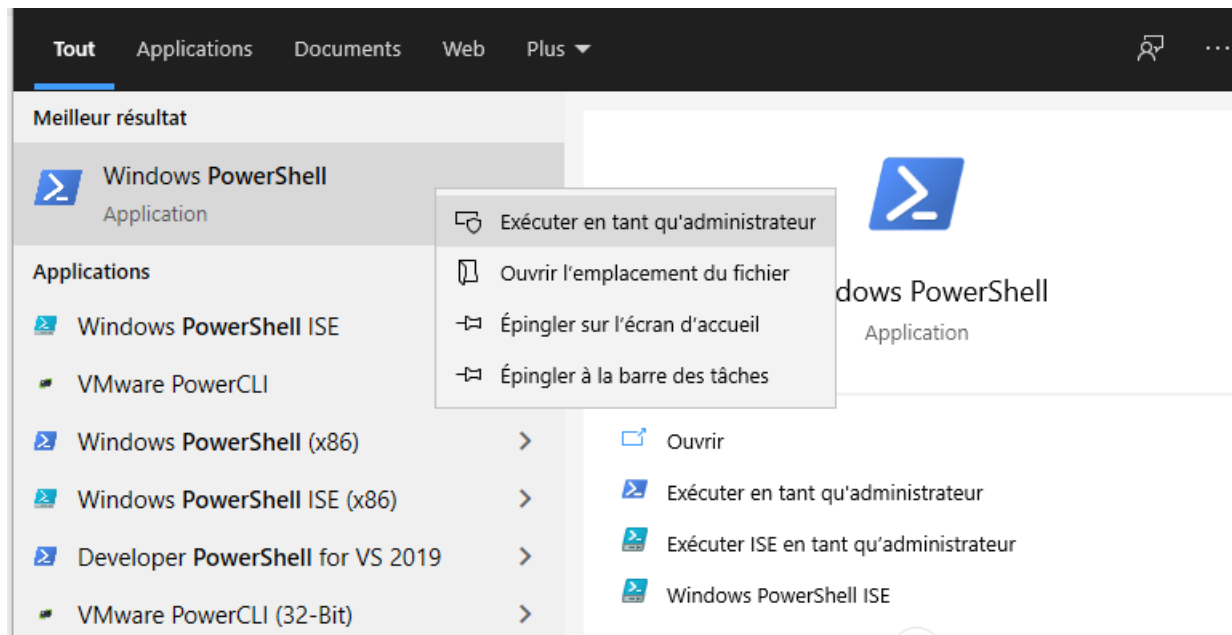


FIGURE 3 – Lancer PowerShell avec en tant qu'administrateur

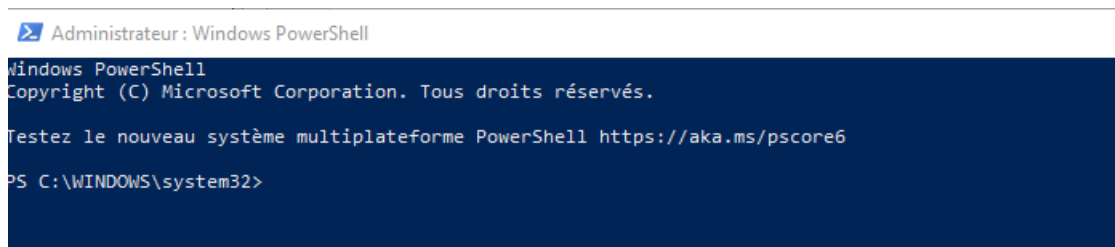


FIGURE 4 – Lancer PowerShell avec en tant qu'administrateur

mande sur l'ordinateur local. Notez toutefois que toutes les autres commandes exécutées à partir de cette instance de la console PowerShell s'exécutent elles aussi avec des privilèges élevés.

1.3 Quelle est la version de mon instance PowerShell ?

Dans PowerShell, il existe plusieurs variables automatiques qui stockent des informations d'état. L'une de ces variables est `$PSVersionTable`, qui contient une table de hachage pouvant être utilisée pour afficher des informations utiles sur la version de PowerShell :

```
1 c:\...\system32>$PSVersionTable
```

1.4 Installation de la version PowerShell Core

Pour nos exercices, nous allons utiliser la version PowerShell Core disponible sur GitHub à l'adresse : <https://github.com/PowerShell/PowerShell/>.

- Dans la page Web, descendez jusqu'à Get PowerShell
- Sélectionnez Windows(x64) et sur la colonne Download (LTS), cliquez sur .msi, comme à la figure 15.
- Le fichier PowerShell-7.0.5-win-x64.msi devrait être proposé pour l'enregistrement comme à la figure 16.
- Enregistrez le fichier à un emplacement de votre choix. Par exemple Téléchargement.
- Après l'enregistrement, procédez à son installation.

Get PowerShell

You can download and install a PowerShell packag

Supported Platform	Download (LTS)
Windows (x64)	.msi

FIGURE 5 – Version LTS

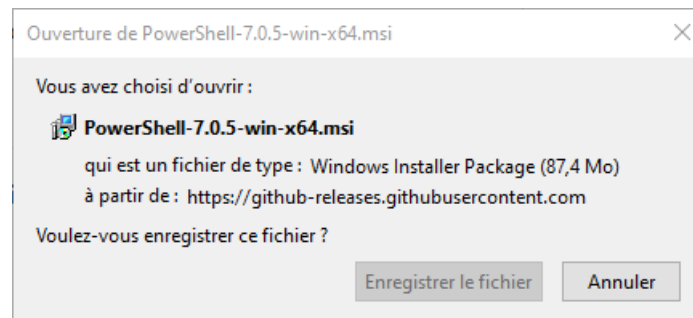


FIGURE 6 – Enregistrer le fichier

- Installer PowerShell 7-64 Setep avec les options par défauts.
- **À l'avenir, vous devrez taper la commande `pwsh` pour lancer PowerShell Core, comme dans les figure 17 et choisir `run as administrator` figure 18.**

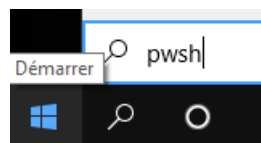


FIGURE 7 – Lancer PowerShell

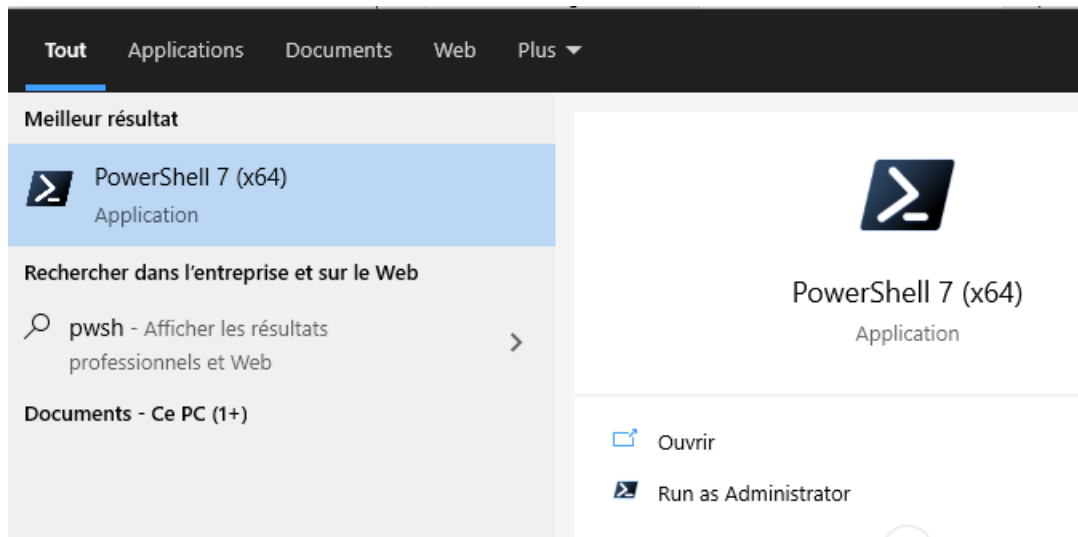


FIGURE 8 – PowerShell 7(x64)

2 Exécuter les commandes du cours

- 1) Récupérer le fichier [CmdPowerShellCoursSE.md](#)¹ depuis le dépôt du cours sur GitHub.
- 2) Pour ce faire, taper la commandes Git suivante si vous n'avez pas le dépôt :

```
2 git clone git@github.com:jpduchesneauCegep/420-W12-SF-4393.git
```

ou si vous avez déjà le dépôt sur votre poste :

```
3 git pull
```

- 3) Ce commande vas récupérer le dépôt Git [420-W12-SF-4393](#) et créer un répertoire de ce nom à l'emplacement ou vous avez appelé la commande.
- 4) Ouvrir le fichier [CmdPowerShellCoursSE.md](#) qui est dans votre nouveau dossier, au sein du dossier PowerShell. Testez les commandes qui y sont.
- 5) Vous pouvez utiliser Visual Studio Code pour lire vos fichier md.

3 Manipulation des dossiers et fichiers

- 1) Pour savoir où vous vous situez, utiliser la commande [Get-Location](#).
- 2) Créer la structure suivante, figure 9 à l'aide des commandes PowerShell.

1. Les fichier portant l'extension md font référence à un format appelé Markdown Documentation File. Vous pouvez en savoir plus sur ces fichiers en allant sur ce lien <https://commentouvrir.com/extension/md>

Commandes de base :

```
4 PS C:\Users\admin>New-Item -type Directory -Name {NomProjet} -Path {Chemin}
```

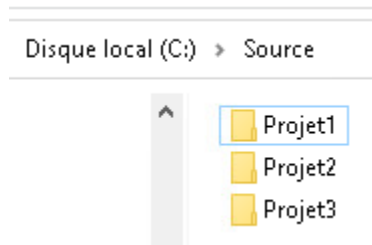
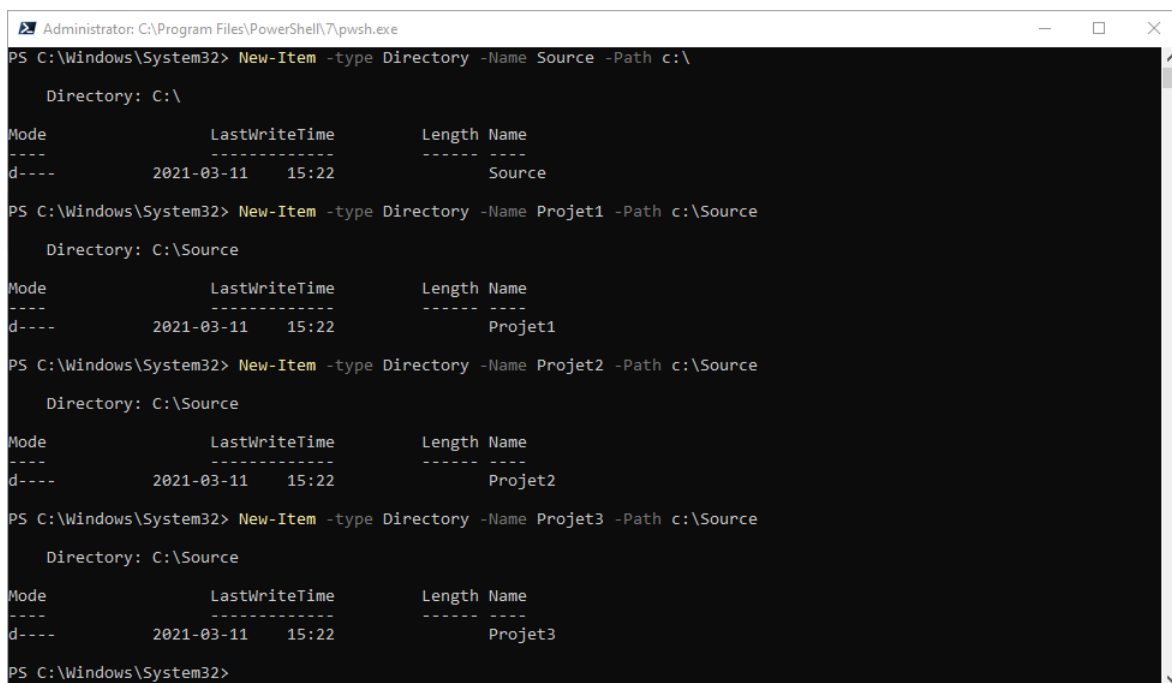


FIGURE 9 – Structure à créer avec PowerShell

Pour y arriver, vous avez probablement tapé quatre fois la commande `New-Item` comme dans la figure 10.



```
Administrator: C:\Program Files\PowerShell\7\pwsh.exe
PS C:\Windows\System32> New-Item -type Directory -Name Source -Path c:\
Directory: C:\
Mode                LastWriteTime         Length Name
-----
d-----         2021-03-11   15:22             Source

PS C:\Windows\System32> New-Item -type Directory -Name Projet1 -Path c:\Source
Directory: C:\Source
Mode                LastWriteTime         Length Name
-----
d-----         2021-03-11   15:22             Projet1

PS C:\Windows\System32> New-Item -type Directory -Name Projet2 -Path c:\Source
Directory: C:\Source
Mode                LastWriteTime         Length Name
-----
d-----         2021-03-11   15:22             Projet2

PS C:\Windows\System32> New-Item -type Directory -Name Projet3 -Path c:\Source
Directory: C:\Source
Mode                LastWriteTime         Length Name
-----
d-----         2021-03-11   15:22             Projet3

PS C:\Windows\System32>
```

FIGURE 10 – Commandes répétitives

Question : Comment auriez-vous pu améliorer cette opération à partir de des notions d'algorithmie apprises dans votre cours de d'introduction à la programmation ?

- 3) Tapez la commande suivante pour afficher votre dossier source :

```
1 Get-ChildItem -Path c:\source
```

- 4) Toujours avec la commande `New-Item`, créez un fichier test.txt dans chacun des répertoires des projet.
- Chacun de ces fichiers doit contenir la chaîne de caractère : "PowerShell"
- 5) Trouver la commande pouvant afficher de façon récursive vos dossiers et vos fichiers du répertoire `Source`.

4 Utilisation de Visual Studio Code pour PowerShell

Comme nous l'avons précisé dans le cours, Microsoft a presque arrêté de développer PowerShell ISE et recommande d'utiliser à la place un outil plus puissant, pratique, flexible et gratuit - Visual Studio Code (VS Code) ; de plus, Visual Studio Code est multiplate-forme donc est aussi disponible sur Mac et Linux et intègre le système de gestion de code Git.

Visual Studio Code est déjà présent sur votre machine virtuelle, par contre, il n'a pas été paramétré pour travailler avec PowerShell.

Pour votre poste personnel, vous pouvez télécharger VSCode gratuitement en suivant ce lien : <https://code.visualstudio.com/>. Téléchargez le fichier d'installation VSCodeSetup-x64 (environ 53 Mo) et exécutez-le.

Pour les exercices du cours, nous vous recommandons d'utiliser la machine virtuelle qui vous est fournie. De cette façon, vous évitez de faire une manœuvre qui pourrait nuire à votre propre PC.

- 1) Ouvrez visual studio code sur la machine virtuelle.
- 2) A gauche, sélectionnez l'icône des extensions comme à la figure 11.
- 3) Dans la nouvelle fenêtre, vous avez un champ "Search Extensions in Marketplace". Tapez `PowerShell` dans ce champ.
- 4) Cliquez install sur l'extension PowerShell proposé comme à la figure 12
- 5) Après l'installation, un message vous indique que vous avez la version de PowerShell (7.0.6) et que la version courante est la version 7.1.3. Il vous demande si vous voulez mettre à jour vers cette version ? Répondez `Yes`.
- 6) Procédez à l'installation de la mise à jour de PowerShell en cliquant sur `Next` et en suivant les étapes. Vous utilisez les paramètres par défaut qui vous sont proposées pour l'installation de la mise à jour.

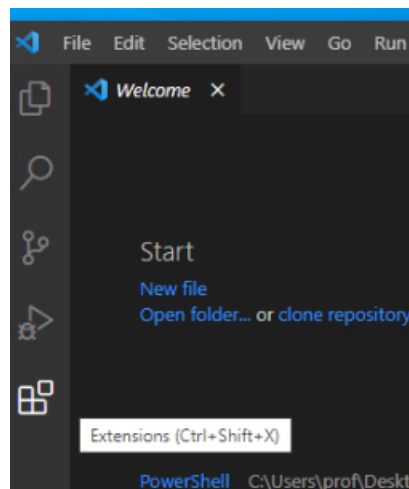


FIGURE 11 – Extensions des Visual Studio Code

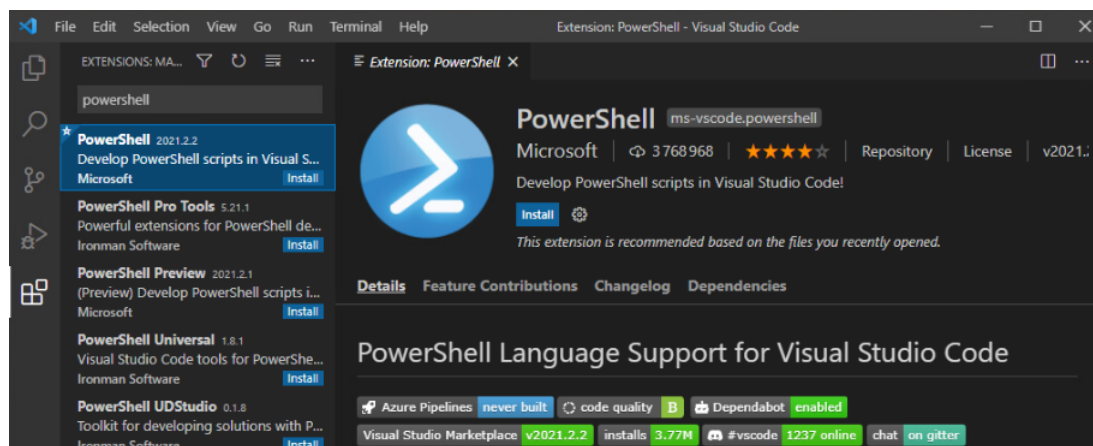


FIGURE 12 – Extension PowerShell

- 7) Bien sur, vous autoriser cette application à apporter des modifications à votre appareil.
- 8) A la fin, cliquez sur [Finish](#)
- 9) Vous devriez avoir la version 7.1 affichée en bas à droite de Visual Studio Code, voir la figure, voir la figure 13. Vous devriez avoir la version 7.1 affichée en bas à droite de Visual Studio Code, voir la figure

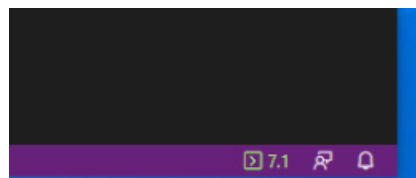


FIGURE 13 – Version PowerShell

- 10) Pour le rendre plus pratique, vous pouvez configurer les paramètres d'interface VSCode suivants

(une icône d'engrenage dans le coin inférieur gauche) à Color Theme = PowerShell ISE voir figure 14.

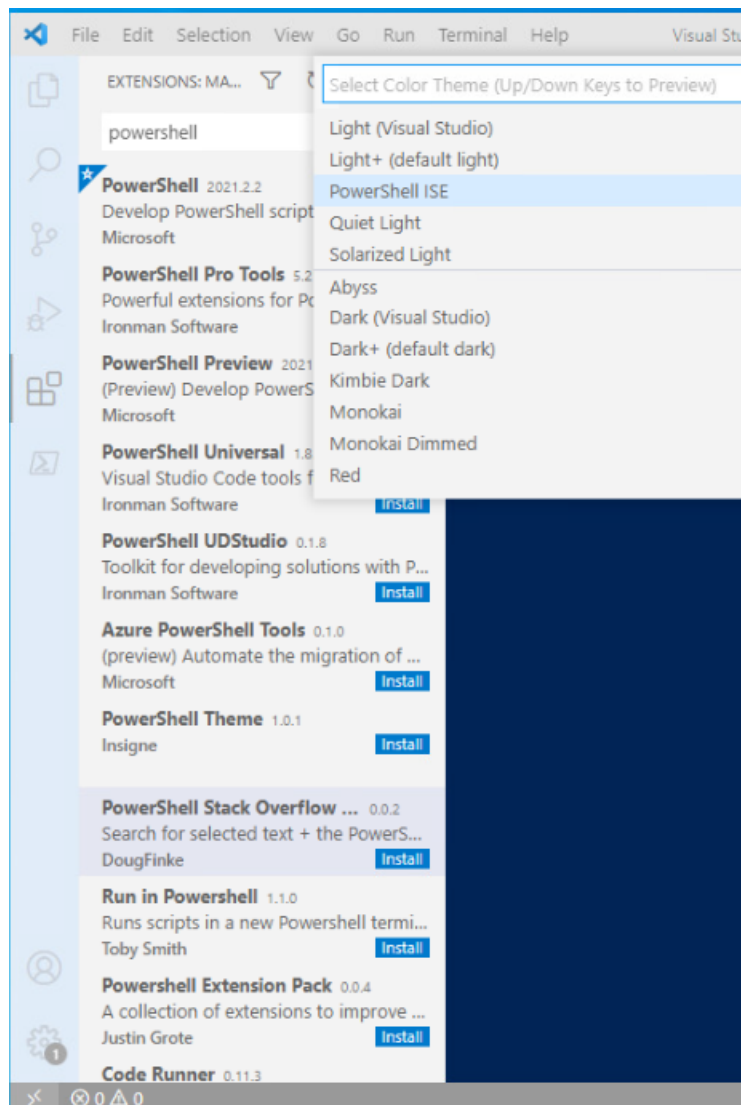


FIGURE 14 – Changer le thème

- 11) Si vous planifiez uniquement l'utilisation de VS Code pour les scripts PowerShell, vous pouvez masquer la Console de débogage, car elle n'est pas utilisée par l'extension PowerShell. Pour ce faire, cliquez avec le bouton droit sur Console de débogage puis cliquez sur la coche pour la masquer.
- 12) Il est possible que VS Code vous propose des mises à jour en bas à droite sur l'icône d'engrenage. N'hésitez pas à faire ces mises à jour.

Pour en savoir d'avantage sur l'utilisation de PowerShell dans VS Code, je vous invite à lire : <https://docs.microsoft.com/fr-fr/powershell/scripting/dev-cross-plat/vscode/how-to-replicate-the-ise-environment?view=powershell-7.1>

5 Mon premier script PowerShell

Passer des one-liners PowerShell aux scripts est bien plus simple que ça n'en a l'air. Un script n'est rien de plus qu'un ensemble de commandes identiques ou similaires que vous exécutez de manière interactive dans la console PowerShell, sauf que ces commandes sont enregistrées dans un fichier .PS1.

- 1) Cliquez sur Ctrl+N, ou file -> New File pour créer un nouveau fichier.
- 2) Sauvegarder ce fichier dans Documents -> PowerShell -> Script avec comme nom : **MonPremierScript.ps1** (Figure 5)

Attention l'extension permet à VisualStudio de reconnaître le langage de programmation utilisé et de ce fait, d'utiliser la bonne extension. Donc, c'est une bonne habitude de toujours nommer vos fichiers avant de débiter.

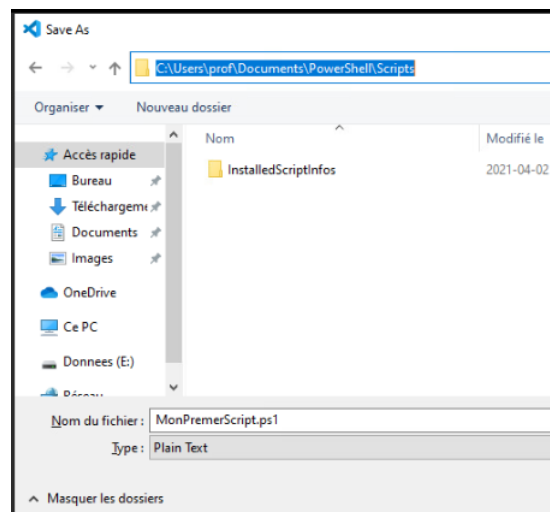


FIGURE 15 – Sauvegarde de mon premier script

- 3) Vous devriez maintenant avoir l'affichage de la figure 16 avec votre fichier en haut et le terminal PowerShell en bas.
- 4) Il existe une alternative pour choisir votre langage : VSCode prend en charge de nombreux langages de shell et de programmation. Pour basculer entre eux, appuyez sur F1. Dans la barre qui apparaît, tapez Changer le mode de langue et sélectionnez la syntaxe de langue que vous souhaitez utiliser. Sélectionnez PowerShell et l'icône du fichier ouvert dans l'onglet actif deviendra celle du PS.
- 5) Vous avez tout ce qu'il faut pour travailler.
- 6) Tapez une commande simple pour afficher la liste des processus en cours d'exécution : **Get-Process**.
 - Vous pouvez la taper directement dans le terminal.
 - Ou encore, la taper dans votre script en haut et le sélectionnez et taper F8.
 - Testez les deux façons. Elles vous donnent le même résultat.(voir figure 17)

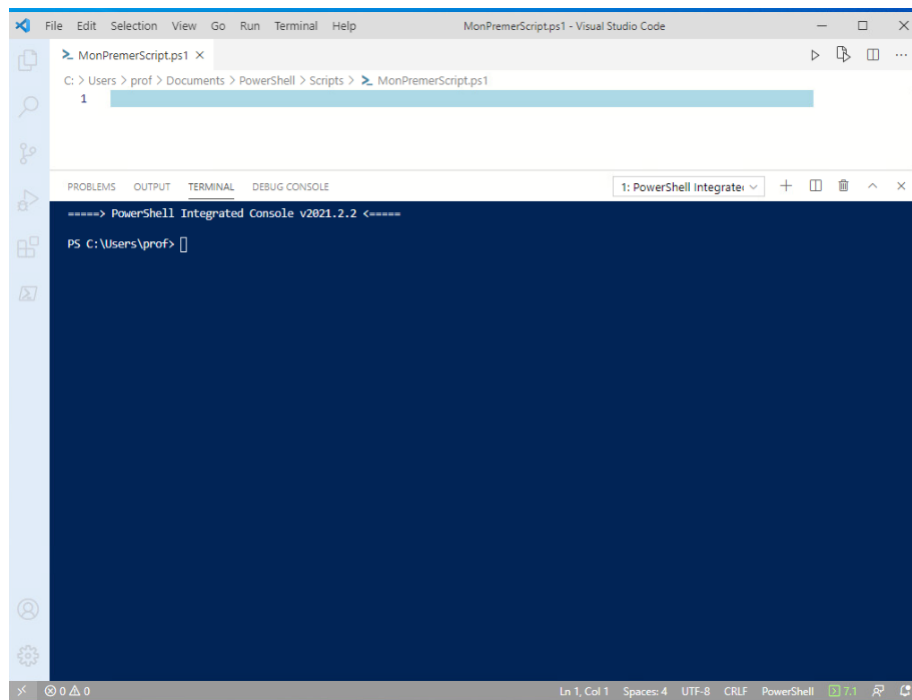


FIGURE 16 – Votre espace de travail

- Bien sûr, ce qui nous intéresse c'est d'écrire des Scripts ; donc, d'écrire dans la partie du haut du code, des variables, des structures de contrôles, des fonctions, etc. qui vous permettront de gagner en temps et en possibilités.

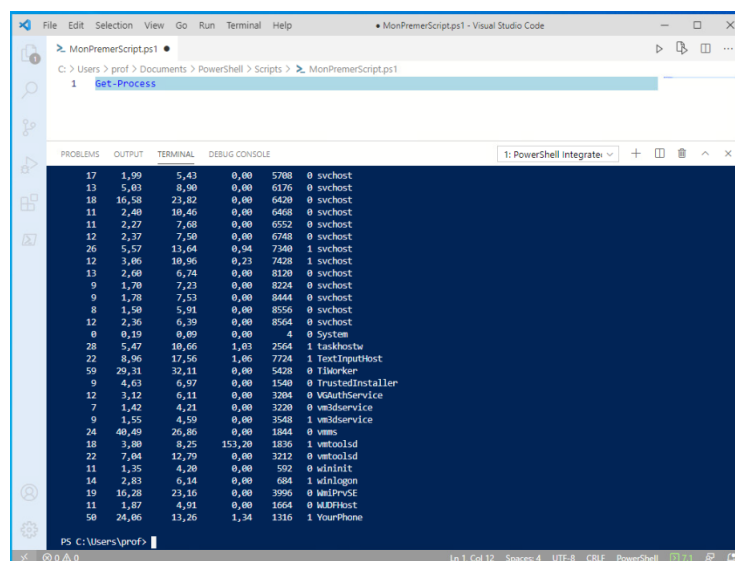


FIGURE 17 – Get-Process

- 7) Comme vous pouvez le voir, la technologie IntelliSense prend en charge l'indication automatique de la commande que vous choisissez en appuyant sur Tab et l'aide intégrée sur les paramètres d'applet de commande disponible.

5.1 Testons l'outil VS Code avec du code

1) Tapez le code suivant :

```

5      # Mon premier script
6      # Version 1.0
7      # [Votre nom]
8      # Avril 2021
9
10     # Déclaration de variables :
11     $date= Get-Date
12     $nomPC = Get-Content env:COMPUTERNAME
13     $ligne="↵
        -----↵
        "
14     $ligneDouble = "↵
        =====↵
        "
15
16     # Programme :
17     Clear-Host
18     Write-Host $ligneDouble
19     Write-Host "Exécution du Script le $($date)"
20     Write-Host $ligne
21     Write-Output "Le nom de l'ordinateur est : $($nomPC)"
22     Write-Host $ligne
23     Get-CimInstance -Query 'Select * from Win32_BIOS'
24     Write-Host $ligne
25     Write-Output "Adresses IPv4"
26     Get-NetIPAddress -AddressFamily IPv4 | Sort-Object -Property ↵
        InterfaceIndex | Format-Table
27     Write-Host $ligneDouble

```

2) Pour comprendre Get-Instance vous pouvez lire : <https://docs.microsoft.com/fr-fr/powershell/scripting/learn/ps101/07-working-with-wmi?view=powershell-7.1>

3) Dans le terminal, déplacez-vous dans le répertoire de votre script et exécutez-le.

```

1      set-location c:\Users\Documents\PowerShell\Script
2      .\MonPremierScript.ps1

```

5.2 Autoriser l'exécution de scripts

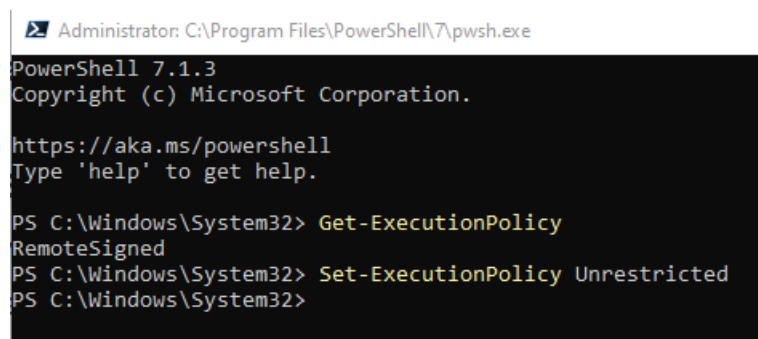
Comme précisé dans le cours, les scripts sont écrits dans VS Code, mais vous devez les exécuter sur votre console PowerShell. Par défaut, PowerShell est configuré en mode "Restricted". Nous allons le basculer en mode "Unrestricted" pour faciliter sa prise en main.

Il est possible d'indiquer à PowerShell plusieurs modes d'exécution. Chaque mode implique un certain comportement :

- **Restricted** : aucun script ne peut être exécuté. PowerShell est utilisable uniquement en mode interactif.
- **AllSigned** : seuls les scripts signés peuvent être exécutés.
- **RemoteSigned** : les scripts téléchargés depuis Internet doivent être signés pour être exécutés. Les scripts présents sur votre poste de travail ne sont pas concernés et peuvent être exécutés.
- **Unrestricted** : pas de restrictions. Les scripts peuvent être exécutés.

Bien sûr, dans un environnement de production, vous devriez utiliser le mode "AllSigned".

- 1) Si vous souhaitez connaître le mode actuellement utilisé, lancez la cmdlet [Get-ExecutionPolicy](#)
- 2) Pour changer de mode, il suffit d'utiliser la cmdlet [Set-ExecutionPolicy](#) [Les modes souhaités] (voir la figure 18).



```
Administrator: C:\Program Files\PowerShell\7\pwsh.exe
PowerShell 7.1.3
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.

PS C:\Windows\System32> Get-ExecutionPolicy
RemoteSigned
PS C:\Windows\System32> Set-ExecutionPolicy Unrestricted
PS C:\Windows\System32>
```

FIGURE 18 – Set-ExecutionPolicy

À partir de maintenant, vous pouvez exécuter n'importe quel script PowerShell(.ps1)

5.3 Création d'un usager avec PowerShell

Dans ce script, vous allez pouvoir créer un usager avec un script.

- 1) Tapez le code suivant :

```
29      # Script de création d'un usager local avec les droits ←
      administrateur
30      # Version 1.0
31      # Auteur [Votre nom]
```

```
32     # Date
33
34     # Création et affectation des variables
35     $nomUsager = "usager1"
36     $nomComplet = "Usager test"
37     $dateExpires = (get-date).AddYears(1)
38     $Description = "Test avec PowerShell"
39     $motPasse = "Soleil01"
40     $groupe = "Administrateurs"
41
42
43     New-LocalUser -Name $nomUsager -AccountExpires $dateExpires -<
        Description $Description -FullName $nomComplet -NoPassword
44     Add-LocalGroupMember -name $groupe -Member $nomUsager
```

- 2) Dans le terminal, déplacez-vous dans le répertoire de votre script et exécutez-le.
- 3) Ouvrez la fenêtre "Gestion de l'ordinateur" et vérifiez la présence de l'utilisateur créé.
- 4) Supprimer l'utilisateur du groupe administrateurs avec la commande suivante :

```
1 Remove-LocalGroupMember -name Administrateurs -Member<
    usager1
```

- 5) Vérifier dans "Gestion de l'ordinateur" les modifications apportées.

6 Références intéressantes

- PowerShell 101 écrit par Mike F Robbins. <https://docs.microsoft.com/fr-fr/powershell/scripting/learn/ps101/00-introduction?view=powershell-7.1>
- Pour mettre à niveau votre version vers Windows PowerShell 5.1, consultez [Mise à niveau des instances Windows PowerShell existantes](#).
- Pour installer la dernière version de PowerShell, consultez [Installation de PowerShell](#).
- Pour installer PowerShell sur Linux, consultez [Installation de PowerShell sur Linux](#).

Sommaire

1	Prise en main de PowerShell	1
1.1	Où trouver PowerShell ?	1
1.2	Comment lancer PowerShell	1
1.3	Quelle est la version de mon instance PowerShell ?	3
1.4	Installation de la version PowerShell Core	3
2	Exécuter les commandes du cours	5
3	Manipulation des dossiers et fichiers	5
4	Utilisation de Visual Studio Code pour PowerShell	7
5	Mon premier script PowerShell	10
5.1	Testons l'outils VS Code avec du code	12
5.2	Autoriser l'exécution de scripts	13
5.3	Création d'un usager avec PowerShell	13
6	Références intéressantes	14
	Sommaire	15
	Références	16

Ce document a été écrit avec LaTeX.

Cette oeuvre, création, site ou texte est sous licence Creative Commons Attribution - Pas d'Utilisation commerciale - Partage dans les Mêmes Conditions 4.0 International. Pour accéder à une copie de cette licence, merci de vous rendre à l'adresse suivante <http://creativecommons.org/licenses/by-nc-sa/4.0/> ou envoyez un courrier à Creative Commons, 444 Castro Street, Suite 900, Mountain View, California, 94041, USA.