

4.9 Les commentaires

La plupart des scripts commence avec des commentaires qui expliquent ce que fait le script, comment il le fait, comment il faut s'en servir etc ... PS supporte 2 types de commentaires :

- *Commentaire d'une seule ligne* qui commence par #. Tout ce qui se trouve après le # n'est pas interprété par PS. *Remarque* : vous pouvez créer des chaînes de caractères contenant des # ; ce caractère ne sera pas interprété comme un commentaire.

```
$myVar = "$env:computername" #Get the computer name
```

- *Commentaires sur plusieurs lignes* qui débute avec <# et qui se termine avec #> :

```
<# -----
ScriptName: EvaluateComp.ps1
Description: This script checks the working environment of a computer
to determine issues with drive space, network connections, etc.
----- #>
```

Tous les scripts que vous créez devraient contenir les informations suivantes :

- Quand le script a été créé et modifié pour la dernière fois ;
- Qui a créé le script (le nom seul n'est pas forcément significatif !!!) ;
- Comment contacter l'auteur ;
- En quoi le script est utile ;
- Est-ce qu'une sortie est prévue pour le script, où et sous quelle forme ?
- Essayez dans la mesure du possible d'écrire vos commentaires et algorithmes en anglais ; de cette façon, ils seront compréhensibles par tout le monde...

Exemple de commentaires assez complet pour un script :

```
<#
.AUTHOR
    Denis Pallez
    Associate Professor at University of Nice - France
    denis.pallez@unice.fr
.SYNOPSIS
    Create a new user account on the local computer.
.DESCRPTION
    This function will create a user account on the local computer.
.PARAMETER Computer
    The NetBIOS name of the computer that you will create the account on.
.PARAMETER User
    The user name of the account that will be created.
.PARAMETER Password
    The password for the account, this must follow password policies
    enforced on the destination computer.
.PARAMETER Description
    A description of what this account will be used for.
.NOTES
    You will need to run this with either UAC disabled or from an elevated
    prompt.
.REQUIRES
    This script requires at least Microsoft PowerShell v1.0
.EXAMPLE
    New-LocalUser -ComputerName MyComputer -User MyUserAccount
                -Password MyP@ssw0rd -Description "Account."
Description
-----
    Creates a user named MyUserAccount on MyComputer.
```

```
.LINK  
http://www.google.fr  
#>
```

Exercice 42 Créer un script qui ajoute un numéro de ligne devant chacune des lignes d'un fichier à l'affichage. Indices : récupérer le contenu du fichier et l'envoyer sur un pipe pour y introduire avant un numéro qui s'incrémente ; il est possible d'utiliser les opérateurs `$_` ; `ForEach-Object` (`%{}`). Merci de commenter votre script comme dans l'exemple précédent. Dorénavant, prenez le réflexe de commenter tous vos nouveaux scripts de la même manière. Même si cela vous paraît être une perte de temps au début, vous trouverez cela très utile quand vous irez chercher des solutions sur le net car vous serez très content que l'auteur ait commenté ses scripts ou quand vous relirez vos scripts pour l'examen final !

Exercice 43 Écrivez un script qui supprime les fichiers du répertoire `c:\Temp` qui n'ont pas été accédés depuis plus de 10 jours. Indice : utilisez la cmdlet `Get-Date` pour avoir la date du jour et y ajouter `-10` jours.

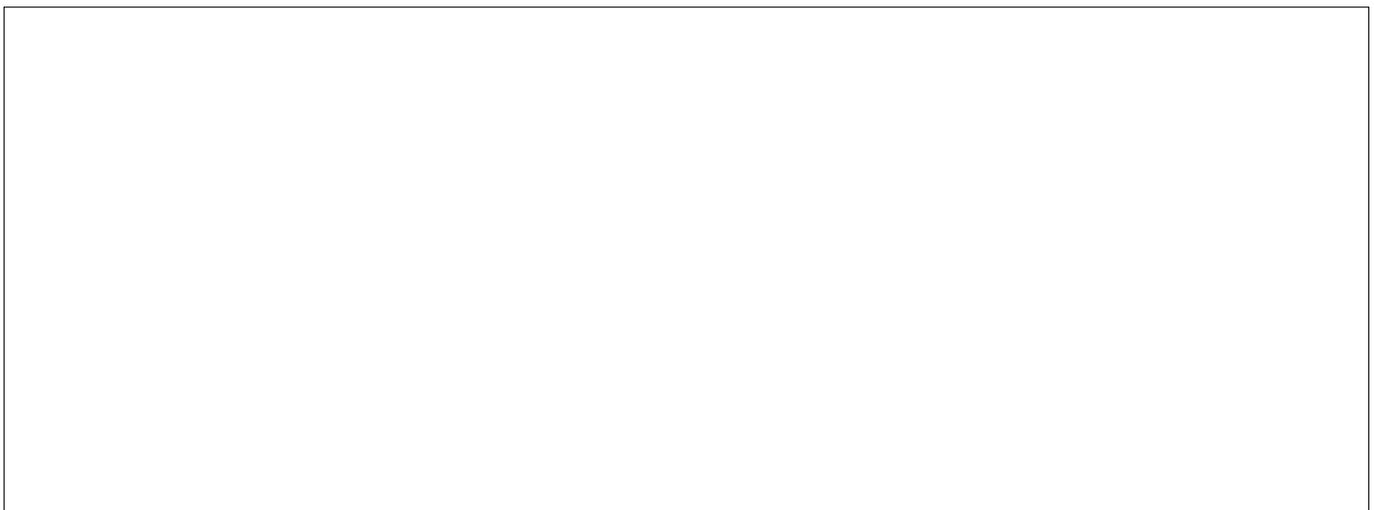
Exercice 44 Écrivez un script qui supprime tous les processus qui ne répondent plus. Afin de tester correctement votre script, il est possible de faire *semblant* de supprimer les scripts qui répondent avec l'option `-whatif`.



Exercice 45 Récupérer la taille de mon répertoire de documents. Voici quelques indices :

1. Récupérer le chemin vers les documents de l'utilisateur courant. Pour cela, il est fortement conseillé d'utiliser la méthode `GetFolderPath` de la classe `System.Environment` de .NET (<http://msdn.microsoft.com/en-us/library/system.environment.aspx>). La syntaxe pour accéder à cette méthode via cette classe en PS est `[Environment]::GetFolderPath()`. Cette méthode demande un argument qui correspond à un type énuméré accessible via la syntaxe `[Environment+SpecialFolder]` suivi de la valeur du type énuméré souhaité.
2. Utiliser également la cmdlet `Measure-Object`.
3. Formater votre sortie en utilisant des Kilo, Méga, Giga, Téra octets et en affichant des valeurs lisibles comme le montre l'écran suivant :

```
PS C:\Users\denis> scripts:\SizeOfFolder.ps1  
13,42 MegaBytes
```



Exercice 46 Écrivez une fonction qui attend qu'une touche au clavier soit saisie. Indice : utilisez la classe `Console` (méthode `ReadKey`) disponible dans .NET.



Exercice 47 Écrivez un script qui supprime tous les sous-répertoires du répertoire courant ou des répertoires passés en paramètre. Pour cela, utilisez la classe `System.IO.Directory` de .NET.

