**Documentation sur la programmation PowerShell**

**Fonction avancée dans PowerShell**

Utilisation de l’attribut **CmdletBinding** et de l’attribut **Parameter**

Voici la fonction avancée la plus simple

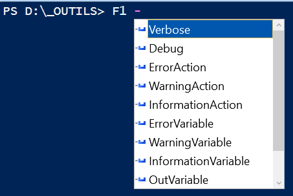
**Function F1 {**

**[CmdletBinding()]Param()**

**}**

Si on enregistre la fonction F1 dans un fichier TEST\_F1.PS1

L’attribut **CmdletBinding** permet d’avoir accès à plusieurs paramètres



L’attribut **CmdletBinding** permet d’utiliser **Write-Verbose** dans une fonction

**Function F2 {**

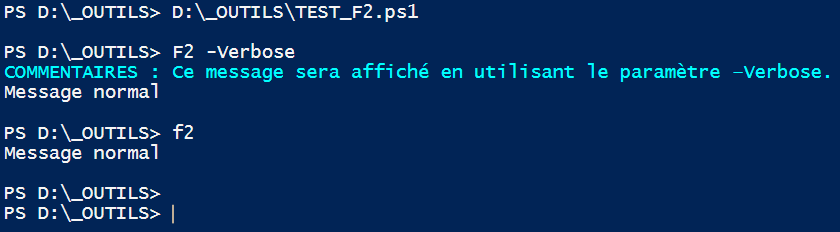
**[CmdletBinding()]Param()**

**Write-Verbose "Ce message sera affiché en utilisant le paramètre –Verbose."**

**Write-Host "Message normal"**

**}**

Si on enregistre la fonction F2 dans un fichier TEST\_F2.PS1



**La fonction F3 utilise un paramètre obligatoire**

**Function F3 {**

**[CmdletBinding()]**

**Param([Parameter(Mandatory=$True)]**

**[String]$Source**

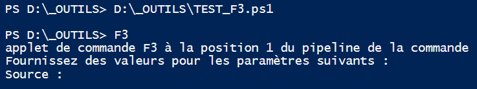
**)**

**Write-Host $Source -ForegroundColor Yellow**

**}**

Si on enregistre la fonction F3 dans un fichier TEST\_F3.PS1

PowerShell nous demande de fournir une valeur pour le paramètre de notre fonction



**La fonction F4 utilise un paramètre pour afficher une fenêtre de confirmation.**

**Function F4 {**

**[CmdletBinding(SupportsShouldProcess)]**

**Param([String]$Fichier)**

**$ConfirmPreference = "Low"**

**$info = "Voulez-vous supprimer par la force ce fichier !!!"**

**If ($PSCmdlet.ShouldContinue($info,"Paramètre -Force"))**

**{**

**Remove-Item $Fichier -Force**

**} Else**

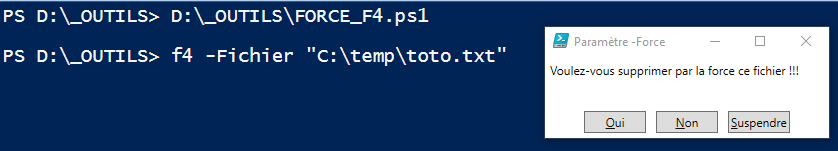
**{**

**Write-Host "Annulation de la suppression du fichier !!!" -ForegroundColor Cyan**

**}**

**}**

Si on enregistre la fonction F4 dans un fichier FORCE\_F4.PS1



**Fonctions avec des variables de type "reference"**

**Exemle 1**

**function carre([ref]$x)**

**{**

**$x.value = $x.value \* $x.value**

**}**

**$nombre = 2**

**Write-Host "Valeur initiale = $nombre" -ForegroundColor Yellow**

**carre([ref]$nombre)**

**Write-Host "Valeur modifiée = $nombre" -ForegroundColor Yellow**

**Exemle 2**

**function double**

**{**

**Param ([ref]$x)**

**$x.value = $x.value \* 2**

**}**

**$nombre = 8**

**Write-Host "Valeur initiale = $nombre" -ForegroundColor Yellow**

**double([ref]$nombre)**

**Write-Host "Valeur modifiée = $nombre" -ForegroundColor Yellow**

**Installation d'un module PowerShell**

Cette section explique où installer un module PowerShell.

L'emplacement du module varie selon l'utilisation

* Pour un utilisateur spécifique
* Pour tous les utilisateurs

Un module est constitué d'une ou plusieurs fonctions.

Si votre script contient plusieurs fonctions (f1, f2, f3, f4, f5) mais que vous voulez rendre disponibles seulement les fonctions f4 et f5, à la fin du module vous devez ajouter les lignes de code suivantes:

**Export-ModuleMember -Function f4**

**Export-ModuleMember -Function f5**

**Installation d'un module pour un "utilisateur spécifique"**

Pour qu'un utilisateur puisse avoir accès à un module qu'il a créé ou qu'il a téléchargé à partir d'un site web comme "Script Center" **https://gallery.technet.microsoft.com/scriptcenter**, il faut installer le module dans un dossier qui est spécifique à l'utilisateur.

**$HOME\Documents\WindowsPowerShell\Modules\MonModule\MonModule.psm1**

IMPORTANT: le nom du dossier **MonModule** et le préfixe du nom du fichier **MonModule**.psm1 doivent être exactement le même, sinon PowerShell ne trouvera pas le module.

**Installation d'un module pour "tous les utilisateurs"**

Pour que tous les utilisateurs puissent avoir accès à un module qu'un administrateur a créé ou téléchargé, il faut installer le module dans le dossier "C:\Program Files".

**$ENV:ProgramFiles\WindowsPowerShell\Modules\MyModule\MyModule.psm1**

IMPORTANT: le nom du dossier **MonModule** et le préfixe du nom du fichier **MonModule**.psm1 doivent être exactement le même, sinon PowerShell ne trouvera pas le module.

**Exemple d'un module PowerShell**

**# Voici le code du fichier EmptyFile.psm1**

**# Permet de créer rapidement un fichier vide en spécifiant sa taille**

**function New-EmptyFile**

**{**

**param( [string]$FilePath,[double]$Size )**

**# Utilisation d'une classe .NET**

**$file = [System.IO.File]::Create($FilePath)**

**$file.SetLength($Size)**

**$file.Close()**

**Get-Item $file.Name**

**}**

**# Voici l'emplacement du fichier EmptyFile.psm1**

**$HOME\Documents\WindowsPowerShell\Modules\EmptyFile\EmptyFile.psm1**

**# Exemple d'utilisation de la fonction New-EmptyFile**

**New-EmptyFile -FilePath c:\\_temp\big\_file.txt -Size 100gb**

**Répertoire : c:\\_temp**

**Mode LastWriteTime Length Name**

**---- ------------- ------ ----**

**-a---- 2021-04-30 08:06 107374182400 big\_file.txt**