

ICT122 –8- Scripts avancé

Objectif

Connaitre les différences entre paramètres et arguments et l'utilisation de *\$args*. Pouvoir définir les attributs des paramètres comme *Mandatory*, *Position* et *ValueFromPipeline*.

Remarque

Il est possible de mettre des « .ps1 » sur Teams, mais dans le cas où les fichiers doivent être envoyés par email, il faut les mettre dans un dossier zippé.

Exercices

1.- Ecrire un script *Write-Numbers.ps1* qui a 2 nombres en paramètre et qui crée un tableau avec les 2 nombres comme limite, puis afficher en parcourant le tableau, tester tous les cas !

```
PS C:\Users\ISI> S:\Scripts\isi-Write-Numbers.ps1 -Min 4 -Max 9
4
5
6
7
8
9
```

2.- Ecrire un script *Read-Numbers.ps1* qui affiche le nombre entré au clavier aussi longtemps que celui-ci est inférieur à la valeur passée en paramètre. On utilise *Read-Host* pour lire un nombre.

```
PS C:\Users\ISI> S:\Scripts\ISI-Read-Numbers.ps1 -NumberMax 33
Le nombre entier à afficher est : 5
5
Le nombre entier à afficher est : 27
27
Le nombre entier à afficher est : 6
6
Le nombre entier à afficher est : 31
31
Le nombre entier à afficher est : 3
3
Le nombre entier à afficher est : 4
4
Le nombre entier à afficher est : 44
```

3- Reprendre le code de *ISI-Test-Function-Date.ps1*(qui est en exemple sur le site) et le modifier pour que la couleur soit passée en paramètre

4.- Ecrire un script *Check-Args.ps1* qui affiche les arguments

```
PS C:\Users\admin> y:\scripts\Check-Args.ps1 toto 3.85 w 6798
toto
3.85
w
6798
```

5.- Ecrire un script *Check-Param.ps1* qui affiche les 2 paramètres, avec un paramètre *Name* qui contient les attributs de paramètres et un autre qui est un nombre *nb*. Tester avec 0, le premier, le deuxième et les deux paramètres

6.- Créer un script *Add-Numbers.ps1* qui additionne les nombres entrés en argument au lancement du script et qui utilise une fonction *Is-Num* pour vérifier que la valeur est numérique

```
PS C:\Users\ISI> S:\Scripts\ISI-Add-Numbers.ps1 3 78 3.25 toto 55.5 z 2
La somme vaut : 141.75
```

7.- Ecrire un script *Create-Group.ps1* qui crée un groupe. Valable pour les exos 7/8/9 : les informations sont passées en paramètres et tous les cas doivent être testés

8.- Ecrire un script *Create-User.ps1* qui crée un utilisateur

9.- Ecrire un script *Add-UserGroup.ps1* qui ajoute un utilisateur dans un groupe