

5 points

Exercices de révision en classe : cours de jeudi 23 mars 2017

1. Vrai ou faux?

	Vrai	Faux
a) Je peux filtrer les objets d'un résultat avec la cmdlet out-file.		X
b) Le foreach permet de parcourir une collection d'objets un par un.	X	
c) La cmdlet sort-object me permet de trier en ordre croissant seulement.		X
d) Les paramètres d'une fonction peuvent être obligatoire.	X	
e) Je suis la cmdlet qui me permet de faire des sélections sur une chaîne de caractère passée dans un pipe. Je suis Select-String.	X	

6 points

2. Écrivez les pipelines de cmdlet pour réaliser les actions suivantes :

3 points

a) Indiquez la commande nécessaire pour tuer tous les processus dont le nom commence par « t », mais en demandant une confirmation pour chacun d'eux.

```
get-process | where-object { $_.name.ToLower().startswith(« t ») } | stop-process -confirm
ou stop-process -name t* -confirm
```

3 points

b) Affichez tous les processus qui ont utilisé plus de 5 secondes de CPU et plus de 1Mo de mémoire vive.

```
get-process | where-object { $_.CPU -gt 5 -and $_.ws -gt 1024 * 1024 }
```

3 points

3. Comment puis-je déclarer une fonction en powershell qui accepte 2 paramètres. Le premier est optionnel et possède la chaîne « -P » par défaut et l'autre de type string qui possède la chaîne « C:\windows » par défaut ?

- a) `function LaFonction([string] $P1 = "-P", [string]$P2="C:\windows")`
- b) `function LaFonction((parameter) [Mandatory=true][int] $P1 = "-P", [string] $P2="c:\windows")`
- c) `function LaFonction(mandatory[int] $P1 = "C:\windows", [string]$P2="-P")`
- d) `function LaFonction([int] [Parameter(Mandatory=$true)] $P1="", [string] $P2="C:\windows")`
- e) `function LaFonction([int] [Parameter=$true] $P1="-P", [string] $P2)`

4 points

4. Soit le script nommé « monscript » et son bout de code suivant :

```
param([string] $Phrase, [System.ConsoleColor] $FG = "Red")
```

Parmi les appels suivants, lesquels sont valides en rapport avec le bout de code précédent ?

- a) `./monscript -Phrase "Allo toi" Blue`
- b) `./monscript "Allo toi"`
- c) `./monscript Yellow -Phrase "Allo toi"`
- d) `./monscript Red "Allo toi"`
- e) `./monscript -Phrase "Allo toi"`

12 points

5. Trouvez les erreurs(s'il y a lieu) dans les définitions de fonctions suivantes :

a) fonction MaFonction ()
`function MaFonction()`

b) MaFonction \$Param1, \$Param2
`function MaFonction ($Param1, $Param2)`

c) fonction (\$Param1, \$Param2 = 0)
`function NomFonction ($Param1, $Param2 = 0)`

d) fonction MaFonction([string] \$Param1 = "Nom")

e) fonction MaFonction([string] parameter ([mandatory=true]) \$Montant, [double] parameter (optional=\$true) \$Taxe = 0.10)
`function MaFonction([string] [Parameter(Mandatory=$true)] $Montant, [double] $Taxe = 0.10)`

10 points

6. Écrivez une fonction nommée « countdown » et qui effectuera un compte à rebours à partir d'une valeur entière passée en paramètre à la fonction. Cette valeur représente le temps en minutes à partir duquel le compte à rebours doit commencer et aura une valeur par défaut de 1 minute. Le paramètre se nomme « start ».

Voici des exemples d'appels :

```
countdown -start 30
countdown 5
countdown
```

On a commencé le script pour vous. Il s'agit de le compléter pour répondre au critère.

```
function countdown (??) ([int] $Start = 1)
{
    $minutes = ?? $minutes = $Start
    $secondes = 0

    write-Host "$minutes : $secondes"
    $minutes--

    for ($min = $minutes; $min -ge 0; $min--)
    {
        for ($sec = 59; $sec -ge 0; $sec--)
        {
            write-Host "$minutes : $secondes"
            sleep 1 # pour vraiment faire attendre 1 seconde!
        }
    }
}
```

}

}