Exercices de révision en classe : cours de lundi 8 Janvier 2018

5 points

1. Vrai ou faux?

|  |  |  |
| --- | --- | --- |
|  | **Vrai** | **Faux** |
| 1. Je peux filtrer les objets d'un résultat avec la cmdlet out-file.
 |  | X |
| 1. Le foreach permet de parcourir une collection d’objets un par un.
 | X |  |
| 1. La cmdlet sort-object me permet de trier en ordre croissant seulement.
 |  | X |
| 1. Les paramètres d'une fonction peuvent être obligatoire.
 | X |  |
| 1. 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 |  |

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

6 points

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

3 points

 \_\_\_\_get-process | where-object { $\_.name.tolower().startswith(« t ») } | stop-process -confirm

ou stop-process -name t\* -confirm

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

3 points

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

1. 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 ?

3 points

1. function LaFonction([string] $P1= ''-P'', [string]$P2=''C:\windows'')
2. function LaFonction((parameter) [Mandatory=true][int] $P1= ''-P'', [string] $P2=''c:\windows'')
3. function LaFonction(mandatory[int] $P1= ''C:\windows'', [string]$P2=''-P'')
4. function LaFonction([int] [Parameter(Mandatory=$true)] $P1='', [string] $P2=''C:\windows'')
5. function LaFonction([int] [Parameter=$true] $P1=''-P'', [string] $P2)

4 points

1. 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 ?

1. ./monscript -Phrase ''Allo toi'' Blue
2. ./monscript ''Allo toi''
3. ./monscript Yellow -Phrase ''Allo toi''
4. ./monscript Red ''Allo toi''
5. ./monscript -Phrase ''Allo toi''

12 points

1. Trouvez les erreurs(s’il y a lieu) dans les définitions de fonctions suivantes :
2. fonction MaFonction ()

function MaFonction()

1. MaFonction $Param1, $Param2

function MaFonction ($Param1, $Param2)

1. function ($Param1, $Param2 = 0)

function NomFonction ($Param1, $Param2 = 0)

1. function MaFonction( [string] $Param1 = ''Nom'')
2. function MaFonction([string] parameter ([mandatory=true]) $Montant, [double] parameter ([optionnal=$true]) $Taxe = 0.10 )

function MaFonction([string] [Parameter(Mandatory=$true)] $Montant, [double] $Taxe = 0.10 )

1. Écrivez une fonction nommée « countdown » et qui effectuera un compte à rebourd 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 rebourd doit commencer et aura une valeur par défaut de 1 minute. Le paramètre se nomme « start ».

10 points

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 attende 1 seconde!

 }

 }

}