S3T6 PowerShell 4

Révisions PS

Attention à autoriser les scripts ou utiliser une VM Windows (locale ou proxmox)

Les indices

Créer un objet :

new-item … -itemtype …
Pour un répertoire, il existe des alias plus simples
Pour un fichier, il est aussi possible d'utiliser une redirection

Ecrire dans un fichier :

a) Avec des redirections (>=simple, >> = ajout),
   - pour du texte simple,
   - il faut gérer sois-même le format de sortie
b) Avec le tube | ,
   - pour des objets ou des tableaux
c) Avec la fonction d'export : export-csv, export au format CSV
   - pour un tableau de textes
   - attention à choisir le séparateur (-delimiter)

La saisie au clavier se fait avec la commande read-host suivie d'un message,
L'affichage à l'écran (et l'écriture dans un fichier, version 1) se fait avec echo ou write-host

1 apéritif

Avec l'explorateur, créer un répertoire /temp/ps4 sur le disque DATA (D: ou E:, selon la salle)
Effectuer les commandes suivantes en powershell dans ce répertoire :

1. Créer un fichier test.txt qui contient la liste des objets du répertoire c:\windows
2. Créer un répertoire scripts et y entrer
3. A l'aide de la command new-item, créer le répertoire docs dans le répertoire parent (sans changer le répertoire courant)
4. Créer un script qui permet de lire une chaîne au clavier et de l'exporter dans un fichier p2.txt de ce répertoire lors de la validation
   - Pour les bons, afficher le nombre de caractères écrits dans le fichier ...
     Vous avez écrit … caractères dans le fichier p2.txt
   - Pour les très bons, donner le nom du fichier en paramètre, s'il est vide on interrompt le script avec un message d'erreur.

TAF : Donner les commandes nécessaires pour faire ces scripts

2 plat 1

boucle sur un fichier existant

1. Créer le fichier texte societe.txt (voir document drive B2_31T3_societe.txt) avec l'explorateur et notepad++.

2. Écrire un script qui recherche l'identifiant de l'utilisateur local dans ce fichier et retrouve son nom et son prénom.
3. Si l'utilisateur local n'est pas trouvé dans le fichier, émettre un message d'avertissement : "utilisateur non enregistré".
4. Indiquer deux tests qui permettent de vérifier le fonctionnement du script.

3 plat 2

Saisie, redirection et export formaté

1. Écrire un script qui permet de faire la saisie d'un nom, d'un prénom, d'un login et d'un mot de passe puis d'afficher le résultat : "$nom enregistré"
2. Modifier ce script pour écrire les données saisies dans un fichier texte : pwd1.txt
3. Modifier ce script pour écrire le résultat dans un fichier pwd2.CSV

[Pour les bons] modifier ce script pour que le programme ajoute des noms au fichier sans écraser les noms précédemment enregistrés

3 dessert

Traduction d'un algo en powershell, boucles, lecture au clavier, affichage

Écrire le jeu des allumettes de Marienbad :
L'ordinateur et le joueur doivent alternativement retirer entre 1 et 3 allumettes d'un pool de 15 au départ, celui qui enlèvera la dernière allumette aura perdu.
Rédiger l'algorithme qui permet de modéliser ce jeu

```
Debut
|       tas = 10        // nbre initial d'allumettes
|       Répéter
|       |       écrire "le tas contient $tas allumettes"
|       |       Répéter
|       |       |       joueur = lire "vous pouvez retirer entre 1 et 3 allumettes ou 'q' pour
                        quitter"
|       |       Tant Que ((joueur<1 ou joueur >3) et joueur <> 'q'
|       |       Si (joueur <> 'q')
|       |       |       tas= tas-joueur
|       |       |       Si (tas < 1)
|       |       |       |       Ecrire "le joueur à perdu"
|       |       |       |       Break           // fin de la boucle
|       |       |       Sinon
|       |       |       |       Si (tas>3)
|       |       |       |       |       ordi = 3
|       |       |       |       Sinon
|       |       |       |       |       ordi = tas-1
|       |       |       |       Fin Si
|       |       |       |       tas= tas - joueur
|       |       |       |       Si (tas<1)
|       |       |       |       |       Ecrire "ordi a perdu"
|       |       |       |       |       Break
|       |       |       |       Fin Si
|       |       |       Fin Si
|       |       Fin Si
|       Tant Que (tas>0 et joueur <> 'q')
|       Ecrire "bye"
Fin.
```

Alternative à l'algo :

```
Debut
|  tas = 10           // nbre initial d'allumettes
|  nbCoups=1          // nombre de coups joués
|  Répéter
|  |  écrire "le tas contient $tas allumettes"
|  |
|  |  Si (tas = 1)
|  |  |  Ecrire "le joueur a perdu en $nbCoups coups"
|  |  |  Break        // fin de la boucle
|  |  Sinon
|  |  |  Répéter
|  |  |  |  joueur = lire "vous pouvez retirer entre 1 et 3 allumettes ou 'q' pour quitter"
|  |  |  Tant Que ((joueur<1 ou joueur >3) et joueur <> 'q'
|  |  |  Si (joueur = 'q')
|  |  |  |  Break             // fin de la boucle
|  |  |  Fin Si
|  |  |  tas= tas-joueur
|  |  |  nbCoups=nbCoups+1
|  |  Fin Si
|  |
|  |  Si (tas=1)
|  |  |  |  Ecrire "joueur a gagné en $nbCoups coups"
|  |  |  |  Break
|  |  |
|  |  Sinon Si (tas>3)
|  |  |  tas= tas-3
|  |  Sinon
|  |  |  tas=1
|  |  Fin Si
|  Tant Que (tas>1 et joueur <> 'q')
|  Ecrire "bye"
Fin.
```

[Pour les très bons] Réfléchir à la simplification de l'algo

Écrire le programme PowerShell qui permet de réaliser ce jeu (avec l'un des deux algos).

[Pour les bons] Ajouter une fonction qui affiche le nombre de I (i maj) correspondant au nombre d'allumettes du tas

```
function ( (int] $tas) {
        ...
}
```

Exemple :
si tas = 15, on obtient : TAS : IIIIIIIIIIIIIII si tas = 6, on obtient : TAS : IIIIII Et voilà.