PowerShell Ateliers

Table of Contents

Préambule : Ces exercices sont tous réalisables avec le contenu du cours et l'aide intégrée de PowerShell, rien d'autre n'est nécessaire hormis votre cerveau.
Ces exercices peuvent tous être complété avec l'IA mais ce n'est pas elle qui passera l'examen a votre place et le jury tout comme vos futurs employeurs ne vous fourniront pas du travail par ce que vous savez faire un prompt correct. Merci de faire l'effort de vous en passer pour cette semaine.

1. Exercices Didactiques en PowerShell

Pour chaque exercice vous devrez livrer :
- Un organigrame
- Un script commenté

1.1. Exercice : Contrôler la connexion réseau

- **Objectif** : Identifier si une adresse IP répond au ping. (IF)
- **Instruction** :
  1. Demander à l'utilisateur d'entrer une adresse IP.
  2. Effectuer une commande `Test-Connection` sur cette adresse.
  3. Utiliser une structure `If/Else` pour afficher si l'adresse est joignable ou non.
- **Indication** : Testez avec `-Quiet` pour obtenir un résultat booléen.

1.2. Exercice : Analyse de réponses

- **Objectif** : Comprendre l'utilisation de `Switch` pour traiter des entrées multiples.
- **Instruction** :
  1. Demander à l'utilisateur de choisir une option parmi "démarrer", "arrêter" ou "redémarrer".
  2. Utiliser une structure `Switch` pour afficher le message approprié pour chaque cas.
- **Indication** : Prévoyez une réponse par défaut pour les choix invalides.

1.3. Exercice : Identifier une plage IP

- **Objectif** : Utiliser `Switch` pour associer une adresse IP à une catégorie (ex. publique ou autre, privée), respecter les informations de la RFC 1918.
- **Instruction** :
  1. Demander à l'utilisateur d'entrer une adresse IP.
  2. En fonction des plages connues (ex. 192.168.x.x, 10.x.x.x), afficher le type d'adresse.
- **Indication** : l'utilisation de regex pourrait aider.

1.4. Exercice : Créer un menu interactif

- **Objectif** : Utiliser `Switch` pour construire un menu permettant de choisir différentes actions.
- **Instruction** :
  1. Afficher un menu avec plusieurs options (par exemple : "Afficher la date", "Lister les fichiers", "Quitter").
  2. L'utilisateur choisit une option, et une action spécifique est exécutée.
  3. Répétez le menu jusqu'à ce que l'utilisateur choisisse de quitter.

1.5. Exercice : Surveillance CPU

- **Objectif** : Créer une boucle pour analyser des données système.
- **Instruction** :
  1. Répéter 5 fois une commande de surveillance CPU.
  2. Afficher les valeurs obtenues pour chaque itération.
  3. Une fois que cela fonctionne avec le CPU modifiez le script pour qu'il affiche également les informations pour la RAM et l'interface réseau. 4.
- **Indication** : Consultez l'aide de la cmdlet `Get-Counter`. Une pause peut etre introduite entre les itérations avec `Start-Sleep`.

1.6. Exercice : Deviner un nombre

- **Objectif** : Maintenir une boucle tant qu'une condition n'est pas remplie.
- **Instruction** :
  1. le script définira aléatoirement un nombre entre 0 et 9.
  2. Afficher ce nombre (sera commenté par la suite, sert pour le debug).
  3. Afficher un texte qui invite le joueur a faire une proposition pour deviner le nombre.
  4. Afficher les deux nombres. (sera commenté par la suite, sert au debug)
  5. Tester si l'utilisateur a correctement deviné. Si il a gagné le script le félicite et quitte, sinon il l'invite a essayer a nouveau.
  6. Répéter jusqu'à ce que la réponse soit correcte.
- **Bonus** :
  1. Afficher le nombre d'essais nécessaire a l'utilisateur pour gagner.
  2. Les saisies incorrectes ne sont pas prise en compte dans le nombre d'essai.
  3. Le script proposera différent mode de jeux a l'utilisateur via un menu : La valeur a deviner doit elle être :
     -comprise entre 0 et 9
     -comprise entre 0 et 50
     -comprise entre 0 et 100

- **Indication** : L'aleatoire en PowerShell : `Get-Random`. Affichez des indices (plus ou moins).

1.7. Exercice : Suivi des services

- **Objectif** : Surveiller l'état d'un service.
- **Instruction** :
  1. Demander à l'utilisateur le nom d'un service.
  2. Utiliser une boucle `While` pour afficher toutes les 5 secondes l'état du service.
  3. Arrêter la boucle si l'état passe à "Stopped".
- **Indication** : Utilisez `Get-Service` pour suivre l'état.

1.8. Exercice : Contrôle de mot de passe

- **Objectif** : Tester une entrée utilisateur jusqu'à ce qu'elle soit valide.
- **Instruction** :
  1. Demander à l'utilisateur un mot de passe prédéfini (ex. "admin123").
  2. Répéter la demande tant que le mot de passe est incorrect.
  3. Afficher un message de validation si la réponse est correcte.
- **bonus** : Trouver comment faire pour que la demande de saisie d'un mot de passe n'affiche rien a l'ecran.

1.9. Exercice : Détection de port ouvert

- **Objectif** : Répéter une commande jusqu'à ce qu'une condition soit remplie.
- **Instruction** :
  1. Demander à l'utilisateur un numéro de port.
  2. Utiliser une commande pour tester si le port est ouvert (ex. `Test-NetConnection`).
  3. Répéter toutes les 3 secondes jusqu'à ce que le port soit ouvert.

1.10. Exercice : Liste de fichiers

- **Objectif** : Parcourir une liste avec `Foreach`.
- **Instruction** :
  1. Créer une liste de 5 noms de fichiers factices.
  2. Pour chaque nom dans la liste, afficher "Traitement de [nom]".
  3. Afficher un message final une fois la boucle terminée.
- **bonus** : Faites en sortes que le script test si le fichier est bien présent ou non dans l'arborescence souhaitée.

1.11. Exercice : Utilisation des adresses MAC

- **Objectif** : Parcourir les résultats d'une commande système.
- **Instruction** :
  1. Récupérer toutes les adresses MAC disponibles (ex. `Get-NetAdapter`).
  2. Pour chaque adaptateur, afficher son nom et son adresse MAC, si il a un adresse IP affichez la et le script indiquera si elle repond au ping ou non.
- **Indication** : Utilisez `Select-Object` pour cibler les propriétés.

2. Exercices scripting

2.1. Audit de fichiers

- **Objectif** : Audit de fichiers
- **Instruction** : Creer un script qui listera les fichiers non-modifié depuis un labs de temps donné.
  Il affichera au moins le nom, le chemin complet, le poids en MO, la date de derniere modification. Leur classement se fera par date puis par poids.

2.2. Configuration IP

- **Objectif** : Configuration IP
- **Instruction** : Realisez un script qui affichera les noms, adresse MAC, statut et description des équipements reseaux.
  Il devra permettre a l'utilisateur de choisir une carte afin de n'afficher que les informations qui la concerne.
  Faites en sortes que le script adapte le serveur DNS configuré en fonction de l'adresse IP de l'interface. (10.0 ou 10.100 selon le site).
- **Bonus** : Le script sera autonome et ne demandera pas de saisie utilisateur, se lancera a chaque demarrage du systeme.
- **Indication** : A tester sur vos VMs uniquement.

2.3. Application

- **Objectif** : Verification de la presence d'une application
- **Instruction** : Realisez un script qui va verifier si VMWare workstaion est present sur la machine
  le test se fait sur le repertoire d'installation.
  le test se fait par rapport a la base de registre.
- **Bonus** : on souhaite en plus verifier si la version est a jour et alerter l'administrateur si ce n'est pas le cas.
- **Indication** : il y'a peut etre une variable qui stocke l'information des chemins des programmes installés ?

2.4. Utilisateurs

- **Objectif** : Audit des utilisateurs
- **Instruction** : Realisez un script qui affichera la liste des utilisateurs puis demandera de choisir entre les utilisateurs actifs ou inactifs.
  Dans un second temps il affichera la liste des utilisateurs correspondant au critere choisi uniquement.
  Afficher la liste des comptes dont le mdp n'expire jamais et les afficher sous forme de warning.

+ Afficher les comtpes ne s'etant pas connectés depuis 1 mois.

- **Bonus** : Le script doit fonctionner aussi bien avec que sans AD.
  Trouver les ordinateurs qui n'ont pas ete utilisé depuis 6 mois dans l'AD.
- **Indication** : le warning en PowerShell est un flux particulier

2.5. Certificat

- **Objectif** : Audit Certificat
- **Instruction** : Realisez un script qui fait la liste des certificats present sur la machine et remonte les certificats qui vont expirer dan le mois.
  Faites en sortes que le message affichez a l'ecran ait un format que l'on pourrait facilement integrer a un mail.

2.6. Systeme de fichier

- **Objectif** : Systeme de fichiers
- **Instruction** : Faire un script qui liste le contenu d'un repertoire, il affichera en bleu les dossiers et on conservera l'affichage normal pour les fichiers.
  Il affichera en plus les fichiers avec l'attribut archives en vert.
- **Bonus** il affichera en rouge le fichier et le repertoire ayant le poid le plus élevé

2.7. Creation de fichier

- **Objectif** : Creation de fichiers
- **Instruction** : Faire un script qui crée des fichiers avec differentes extensions dans un repertoire donné. (ex : desktop de l'utilisateur actuel)
  Les extensions seront : .xlsx, .docx, .jpg, .gif, .ps1.
  Le script demandera a l'utilisateur ne nombre souhaité de fichiers par extensions.
- **Bonus** : Il sera possible de passer ses parametres au lancement du script.
- **Indication** : `$#, $[1-9], $@`

2.8. Range ton bureau

- **Objectif** : Deplacement de fichiers
- **Instruction** : Faire un script qui affiche la liste des extensions des fichiers present sur le bureau de l'utilisateur et sans qu'une extension apparaissent plusieurs fois.
  Le script déplace les fichiers avec differentes extensions dans un repertoire portant le nom de l'extension.
- **Bonus** : Le script proposera une fois terminé de deplacer les repertoires et leurs contenus dans sur un emplacement plus appriopríé pour le stockage de documents que le bureau.

2.9. copie de fichier

- **Objectif** : Copie de fichiers
- **Instruction** : Faire un script qui liste le contenu d'un repertoire renseigné par l'utilisateur.
  Puis copie le contenu de ce repertoire dans un repertoire choisi par l'utilisateur.
  Il permettra de renommer chaque fichier selon un pattern choisi.
  Exemple : renommer des photos en les prefixant d'une indication de lieu ou d'evenement suivi de la date puis du nom du fichier.
- **Bonus** : lorsque le script a terminé la copie et verifié qu'il a bien traité tout les fichiers, il proposera de supprimer les fichiers d'origines.

2.10. Audit Machine

- **Objectif** : Audit de machine sur une plage IP
- **Instruction** : Faire un script en PowerShell uniquement, qui liste les machines présente sur une plage ip.
  Il devra afficher le Nom de la machine, son adresse IP, le nom de l'utilisateur actuellement connecté, pour toute machine allumée.
- **Bonus** : Est il possible d'optimiser son temps de traitement ?
- **Indication** : !!! Ne tester cela que sur votre propre reseau virtuel. !!!

2.11. Gestion Installation imprimante

- **Objectif** : Installation d'imprimante
- **Instruction** :
  1. Réalisez un script qui permet de faire l'installation d'une imprimante.
  2. Modifiez le script pour qu'il demande la saisie d'une adresse IP. Le script contiendra la liste des imprimantes et de leurs IP, ainsi que le chemin pour atteindre le driver a installer et procedera a l'installation de cette imprimante en particulier+
- **Bonus** : Faites en sorte que si une adresse reseau et non IP est fournit au script alors toutes les imprimantes de ce reseau seront installées sur le poste. Externalisez les informations relatives aux imprimantes dans un fichier CSV.

2.12. Pendu

- **Objectif** : Jeu du pendu
- **Instruction** : Créez un jeu du pendu en PowerShell avec les caractéristiques suivantes :
  Le jeu doit choisir un mot aléatoire dans une liste prédéfinie de mots français.
  Le joueur a 6 essais pour deviner le mot.
  À chaque tour, affichez le mot partiellement deviné et les lettres déjà proposées.
  Le jeu doit gérer les entrées invalides (chiffres, plusieurs lettres, etc.).
  À la fin du jeu, affichez un message de victoire ou de défaite.

- **Bonus** : Avec de l'asciiart affichez l'état du jeu a chaque itération.

2.13. Motus

- **Objectif** : Jeu MOTUS
- **Instruction** : Créer une version simplifiée du jeu Motus avec les règles suivantes :
  Le jeu choisit un mot français aléatoire (6 lettres)
  La première lettre du mot est toujours révélée
  Le joueur dispose de 6 tentatives pour trouver le mot
  À chaque tentative :
  Les lettres correctement placées s'affichent en vert
  Les lettres présentes mais mal placées s'affichent en jaune
  Affichage des lettres déjà proposées
  Gestion des entrées invalides

2.14. QCM

- **Objectif** : Creation d'un Quizz
- **Instruction** : Creer un script qui permet de jouer a un quizz.
  Il affichera la question.
  Il affichera une ou plusieurs proposition de réponses.
  Il affichera la réponse, si la réponse est correcte l'affichage sera en vert.
  Il comptera les points.
  A la fin de son exécution il affichera le nombre de questions repondues avec le nombre de bonne reponses.
- **Bonus** : Les questions auront 3 niveaux de difficultés.
  La banque de questions sera externalisé dans un fichier CSV.
  Faites en sortes qu'il permette a X joueurs de s'affronter.
  Faites en sortes qu'il affiche au debut le top 3 des meilleurs scores avec leurs pseudos et la date.
  Ce top sera maintenu a jour durablement et en cas d'égalité le score le plus récent prendra la place du plus ancien.

2.15. Mail to Prenom Nom

- **Objectif** : manipuler des chaines de caracteres
- **Instruction** : Votre RH vous fournit une liste d'adresses emails (fichier mails.txt), vous avez besoin d'en extraire uniquement les prenoms et les noms.
  1. Le fichier source fournit mails.txt, devra garder son integrité.
  2. Realisez un script qui charge le contenu du fichier, puis suite au traitement, les prenoms et noms devront etre stocké dans un fichier PrenomsNoms.txt.
- **Indication** : split, replace, match
- **Bonus** : le traitement doit se faire en une seule étape.

2.16. Random Stagiaires

- **Objectif** : Justice pour les stagiaires
- **Instruction** : Vous etes un formateur vous avez besoin de pouvoir interroger equitablement vos stagiaires. Vous etes donc en quete d'un outil permettant de repondre a ce besoin.
  1. Trouver la commande qui permet de faire un tirage aleatoire. Quelles sont ses options ?
  2. Creer un script pour un tirage aleatoire Vous decidez de donner un numero a chaque stagiaire. Creer un script qui tire un chiffre aleatoirement dans la limite du nombre de stagiaire. Il affichera la phrase "le numero du désigné volontaire d'office est : < numero >".
  3. Vos stagiaires ne retiennent pas leurs numeros. Vous les trouvez suceptible mais decidez donc qu'il serait plus judicieux de pouvoir afficher leur Prenom et leur nom plutot que des numeros, sous la forme : **le volontaire d'office est : < numero › <Prenom Nom**.
  4. Vous n'etes pas satisfait que les stagiaires puissent etre tiré plusieurs fois et pas d'autres. Trouvez un moyen de corriger cette problematique.
  5. Vous n'etes pas satisfait que le script se ferme une fois terminé Trouvez une solution pour qu'il demande a l'utilisateur du script si il souhaite proceder a un nouveau tirage.
  6. Vous vous rendez compte que lorsque vous relancer le script le lendemain matin il repart de zero. Faites en sortes que ce ne soit pas le cas.
- **Indication** : Les tableaux possèdent des index. A l'inverse d'une variable, le contenu d'un fichier ne s'efface pas.