# POWERSHELL

## Afficher le nom de la machine :

Get-ComputerInfo -property « csname »

## Afficher la liste des UOs :

Get-ADOrganizationalUnit -filter \*

## Afficher la liste des Users :

Get-ADUser -filter \*

## Afficher la liste des groupes :

Get-ADGroup -filter \*

## Compter les nombre d’UOs :

(Get-ADOrganizationalUnit -filter \*).count

## Compter le nombre d’users :

(Get-ADUser -filter \*).count

## Compter le nombre d’users désactivés

(Search-ADAccount -AccountDisabled -UserOnly).count

## Utiliser la commande Write-Host

Write-Host « Hello »

## Utiliser la commande Read-Host

$age = Read-Host « Please enter your age”

## Typer une variable :

[INT] $age = 47

[STRING]$test=”test”

[string] : Représente une chaîne de caractères. Vous pouvez l'assigner en utilisant des guillemets simples ou doubles, par exemple : $maChaine = "Hello".

[int] : Représente un nombre entier. Par exemple : $monEntier = 42.

[double] : Représente un nombre à virgule flottante. Par exemple : $monDouble = 3.14.

[bool] : Représente une valeur booléenne (vrai ou faux). Par exemple : $monBool = $true.

[array] : Représente un tableau. Vous pouvez assigner plusieurs valeurs à une variable de tableau. Par exemple : $monTableau = 1, 2, 3.

[hash table] : Représente une table de hachage (association clé-valeur). Par exemple : $maTableDeHachage = @{ "Nom" = "John"; "Âge" = 25 }.

[object] : Représente un objet. Les objets peuvent avoir des propriétés et des méthodes. Par exemple : $monObjet = New-Object PSObject -Property @{ Nom = "Smith"; Age = 30 }.

[datetime] : Représente une date et une heure. Par exemple : $maDate = Get-Date.

[null] : Représente une variable sans valeur. Par exemple : $maVariable = $null.

[script block] : Représente un bloc de script. Il peut être assigné à une variable pour être exécuté ultérieurement. Par exemple : $monBlocDeScript = { Write-Host "Hello, World!" }.

## Créer miniscript qui demande le prénom et l’age :

[string]$prenom=Read-Host "Quel est votre prénom ?"

[INT]$age=Read-Host "Quel est votre age?"

Write-Host " votre prénom est:" $prenom "et votre age est de"$age "ans"

## Ajouter user avec script

#demande du mot de passe

$Password = Read-Host "mot de passe?"-AsSecureString

#demande le nom d'utilisateur

$name = Read-Host "Nom d'utilisateur?"

#demande le prenom

$givenname = Read-Host "prénom?"

#nom de login

$SamAccountName = Read-host "nom utilisateur?"

#UserPrincipalName

$UserPrincipalName = Read-Host "nom du compte@abdasr24.adds"

#nom du groupe

$group = Read-Host "Nom du groupe?"

#demande UO

$UO= Read-host "nom de l'OU dans laquelle placer le compte?"

$proprieteuser = @{

name=$name

SamAccountName = $SamAccountName

GivenName = $givenname

UserPrincipalName = $UserPrincipalName

path = "OU=$UO,DC=abdasr24,DC=adds"

Accountpassword = $Password

}

New-ADUser @proprieteuser

New-ADGroup -Name $group -GroupScope Global

add-ADgroupmember -identity $group -Members $SamAccountName

### algorythmie

if (condition) {résultat}

## exemple :

[INT]$note = Read-Host "donnez votre note"

if ($note -lt 10) {Write-Host "peut mieux faire"}

elseif ($note -lt 12) {Write-Host "en progression"}

elseif ($note -gt 12) {Write-Host "bon travail"} $

## exemple avec création d’user

$username = Read-Host "Entrez le nom de l'utilisateur"

if (Get-ADUser -Filter {SamAccountName -eq $username} -ErrorAction SilentlyContinue) {

Write-Host "L'utilisateur $username existe déjà dans Active Directory."

} else {

$password = Read-Host "Entrez le mot de passe" -AsSecureString

$userParams = @{

SamAccountName = $username

UserPrincipalName = "$username@votredomaine.com" # Mettez votre domaine

Name = "Nom Complet de $username"

GivenName = "Prénom"

Surname = "Nom de Famille"

DisplayName = "Affichage $username"

Description = "Description de $username"

Enabled = $true

AccountPassword = $password

}

# Créer l'utilisateur dans Active Directory

New-ADUser @userParams

Write-Host "L'utilisateur $username a été créé avec succès dans Active Directory."

}

## Utilisation du switch :

[INT]$note = READ-Host "Donnez Votre note"

switch ($note) {

{$\_ -lt 10} {write-host "résultat médiocre !";break}

{$\_ -le 12} {write-host "résultat moyen";break}

{$\_ -le 16} {Write-Host "Bon résultat!";break}

Default {Write-Host "Mention Très bien!"}

}

## Exemple avec création d’user

$username = Read-Host "Entrez le nom de l'utilisateur"

[INT]$role = Read-Host "`n

tapez 1 pour un rôle d'utilisateur `n

tapez 2 pour un rôle d'invité `n

tapez 3 pour un rôle d'administrateur `n"

if (Get-ADUser -Filter {SamAccountName -eq $username} -ErrorAction SilentlyContinue) {

Write-Host "L'utilisateur $username existe déjà dans Active Directory."

} else {

$password = Read-Host "Entrez le mot de passe" -AsSecureString

$userParams = @{

SamAccountName = $username

UserPrincipalName = "$username@votredomaine.com" # Mettez votre domaine

Name = $username

GivenName = "Prénom"

Surname = "Nom de Famille"

DisplayName = "Affichage $username"

Description = "Description de $username"

Enabled = $true

AccountPassword = $password

}

# Créer l'utilisateur dans Active Directory

New-ADUser @userParams

Write-Host "L'utilisateur $username a été créé avec succès dans Active Directory."

}

switch ($role) {

{$\_ -eq 1} {Add-ADGroupMember -identity "utilisateurs" -members $username}

{$\_ -eq 2} {Add-ADGroupMember -identity "Invités du domaine"-members $username}

{$\_ -eq 3} {Add-ADGroupMember -identity "Administrateurs de l’entreprise"-members $username}

Default {Write-host "tapez un chiffre entre 1 et 3"}

}

Utilisation d’un fichier CSV

Donner une variable pour le chemin du CSV $CSVchemin

$contenuCSV = import-CSV $CSVchemin -délimiteur « , » -encodage utf8

Exemple :

Foreach ($contenuCSV){

}

$CSVFile = "C:\Scripts\AD\_USERS\Medic\_Argile.csv"

$CSVData = Import-CSV -Path $CSVFile -Delimiter ";" -Encoding UTF8

[string]$DC1="abdasr24"

[string]$DC2="adds"

Foreach($Utilisateur in $CSVData){

$UtilisateurNom = $Utilisateur.Nom

$UtilisateurPrenom = $Utilisateur.Prenom

$UtilisateurSamAccount = $Utilisateur.SamAccountName

$UtilisateurOU = $Utilisateur.OU

#$UtilisateurEmail = "$utilisateurSamAccount@abdasr24.adds"

$utilisateurMDP = "azeAZE123-"

# Vérifier la présence de l'utilisateur dans l'AD

if (Get-ADUser -Filter {SamAccountName -eq $UtilisateurSamAccount})

{

Write-Warning "L'identifiant $UtilisateurSamAccount existe déjà dans l'AD"

}

else

{

$userParams = @{

SamAccountName = $UtilisateurSamAccount

UserPrincipalName = "$($utilisateur.SamAccountName)@$DC1.$DC2"

Name = $UtilisateurNom

GivenName = $UtilisateurPrenom

Surname = $UtilisateurNom

DisplayName = "$UtilisateurNom"

Path = "OU=$utilisateurOU,DC=$DC1,DC=$DC2"

Description = "Description de $UtilisateurNom"

Enabled = $true

AccountPassword = ConvertTo-SecureString "azeAZE123-" -AsPlainText -Force

ChangePasswordAtLogon = $true

}

# Créer l'utilisateur dans Active Directory

New-ADUser @userParams

Write-Host "L'utilisateur $username a été créé avec succès dans Active Directory."

}

}

Fonction Stagiaire (utilisationde function)

function Stagiaire {

    [CmdletBinding()]

    param (

        [string]$personneNom,

        [int32]$personneAge

    )

    BEGIN {Write-Verbose "Débur du script." }

    #message de bienvenue

    Process {"Bonjour {0} !  Vous avez {1}ans !" -F $personneNom,$personneAge}

    END {Write-Verbose "Fin du script." }

}

Stagiaire "alice" 35 -Verbose

Stagiaire "Bob" 54 -Verbose