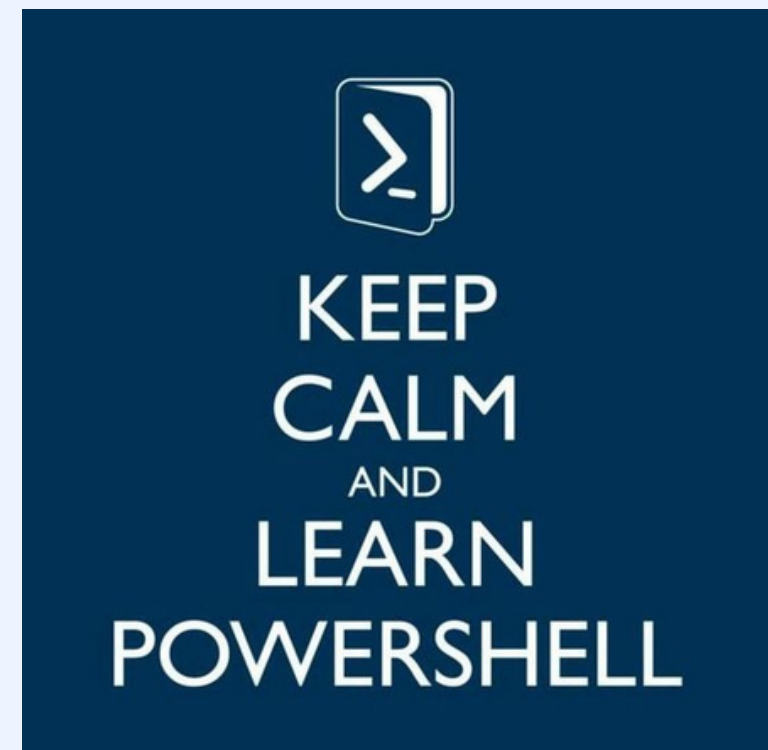




POWERSHELL

Automatisez vos tâches système
avec PowerShell

M. Slimani
2026

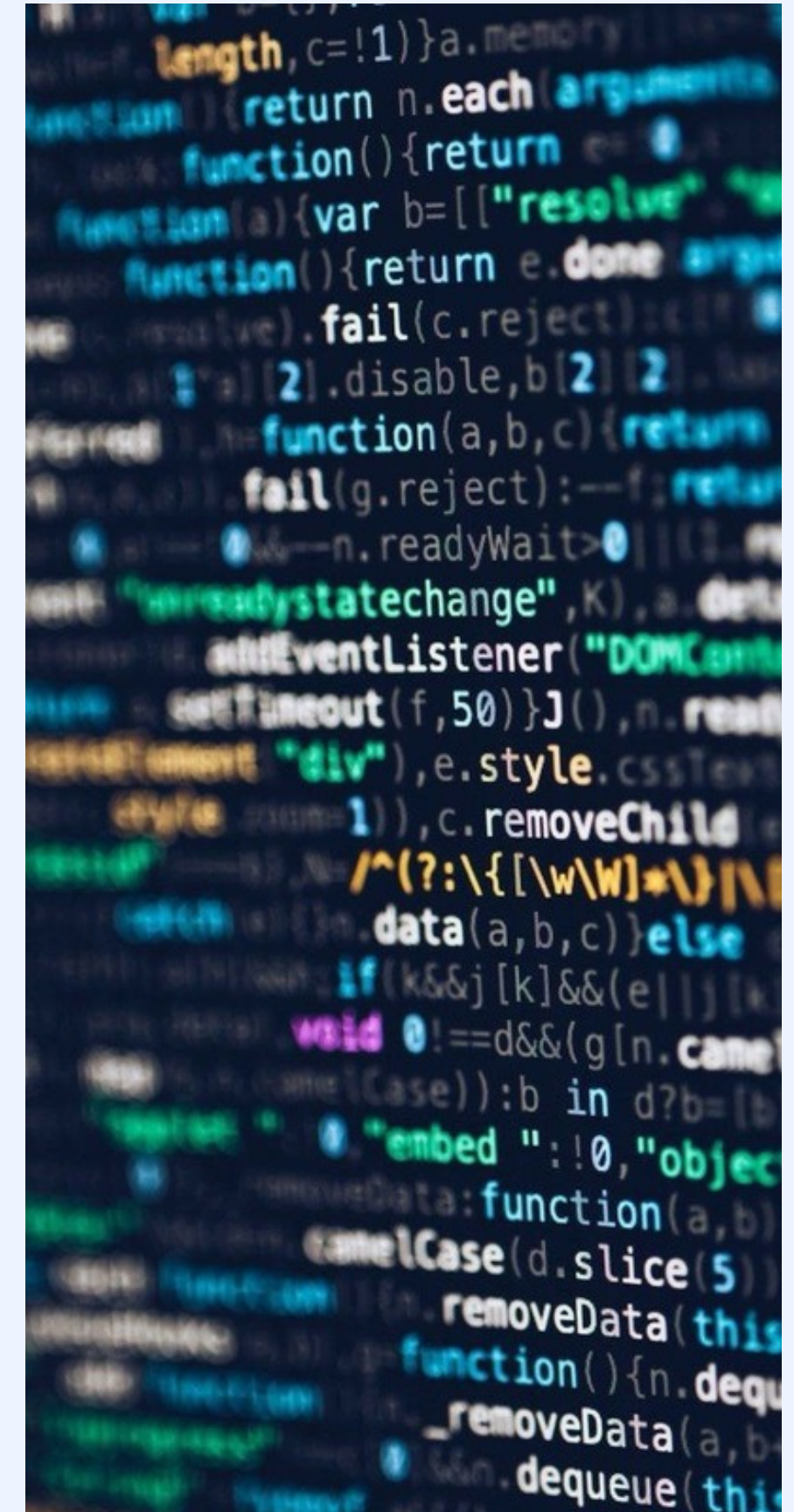


● → Qu'est-ce que PowerShell ?

- PowerShell est un shell de commande et un langage de script développé par Microsoft.
- Basé sur .NET, il permet d'automatiser des tâches d'administration système, de gestion de réseau, d'Active Directory, etc.
- Plus puissant et flexible que l'invite de commandes (cmd).

● → Pourquoi l'utiliser ?

- Automatisation des tâches répétitives
- Gestion centralisée des systèmes Windows
- Intégration avec Active Directory, Exchange, Azure, etc..



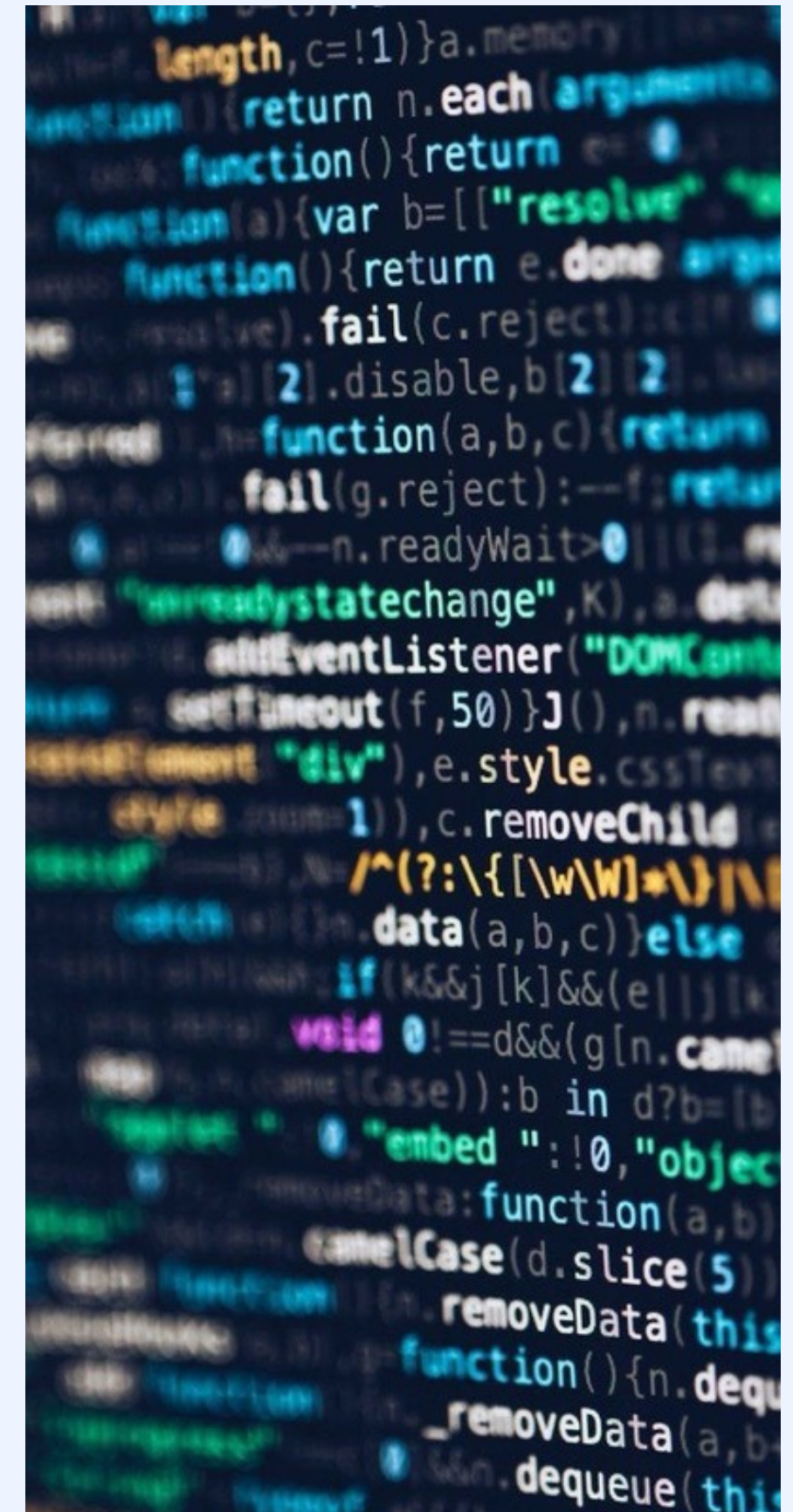
● → L'environnement PowerShell

Comment lancer PowerShell ?

- Windows PowerShell : Version historique (Windows PowerShell 5.1)
- PowerShell 7+ : Version moderne, multiplateforme (Windows, Linux, macOS)

● → Interfaces

- Windows PowerShell ISE : Éditeur intégré (obsolète)
- Visual Studio Code : Éditeur recommandé (extensions PowerShell)





Les cmdlets

Les commandes PowerShell

- Format : Verbe-Nom (ex: Get-Process, Set-Service)
- Obtenir de l'aide : Get-Help Get-Process -Detailed
- Exemples :
 - Get-Date : Obtenir la date système
 - Get-Process : Lister les processus en cours
 - Get-Service : Lister les services Windows

```
Administrateur: Windows-PowerShell  
C:\Users\scott\Desktop> $result = [PSCustomObject] @{  
    Time = Get-Date;  
    Uri = $uri;  
    StatusCode = [int] $request.StatusCode;  
    StatusDescription = HEY $request.StatusDescription;  
    ResponseLength = $request.RawContentLength;  
    TimeTaken = $time;  
}
```



Le pipeline

Enchaîner les commandes

- Le pipeline (|) permet de passer le résultat d'une cmdlet à une autre.

Exemple :

- `Get-Process | Where-Object {$_.CPU -gt 100}`



PowerShell



Variables et types de données

- Déclaration : `$maVariable = "valeur"`
- Types courants :
 - `[int]` : entier
 - `[string]` : chaîne de caractères
 - `[array]` : tableau

Exemple :

- `$aujourdhui = Get-Date -Format "dd-MM-yyyy"`
- `$noms = @("Aya", "Adam", "Mohammed")`

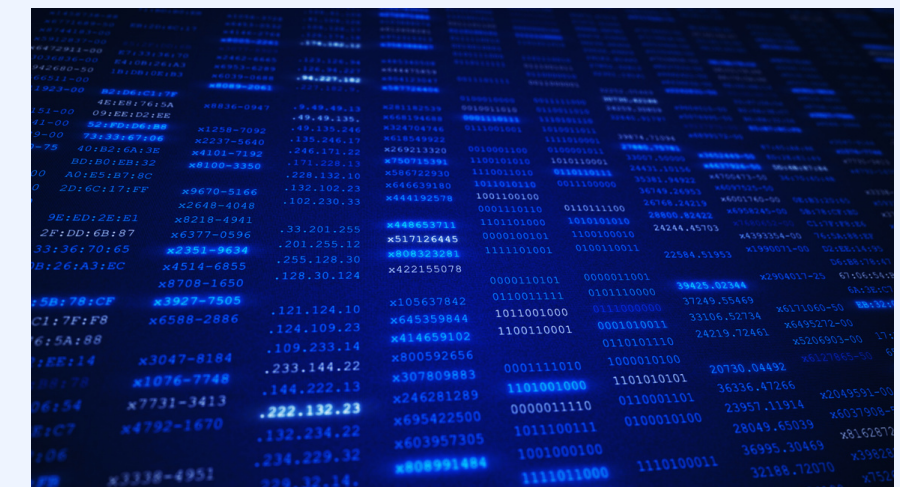




Les conditions

- Conditions : if/elseif/else

```
if (condition1) {  
    # Code si condition1 vraie  
}  
elseif (condition2) {  
    # Code si condition2 vraie  
}  
else {  
    # Code si aucune condition vraie  
}
```





Le switch

- switch : selon le cas

```
switch ($variable) {  
    valeur1 {  
        # Code si $variable = valeur1  
        Pause  
    }  
    valeur2 {  
        # Code si $variable = valeur2  
        Pause  
    }  
    default {  
        # Code si $variable n'est dans la liste  
    }  
}
```

```
switch ($note) {  
    { $_ -ge 16 } { "Très bien" }  
    { $_ -ge 14 } { "Bien" }  
    { $_ -ge 12 } { "Assez bien" }  
    default { "Pas bon" }  
}
```



Les boucles

- Répéter un traitement

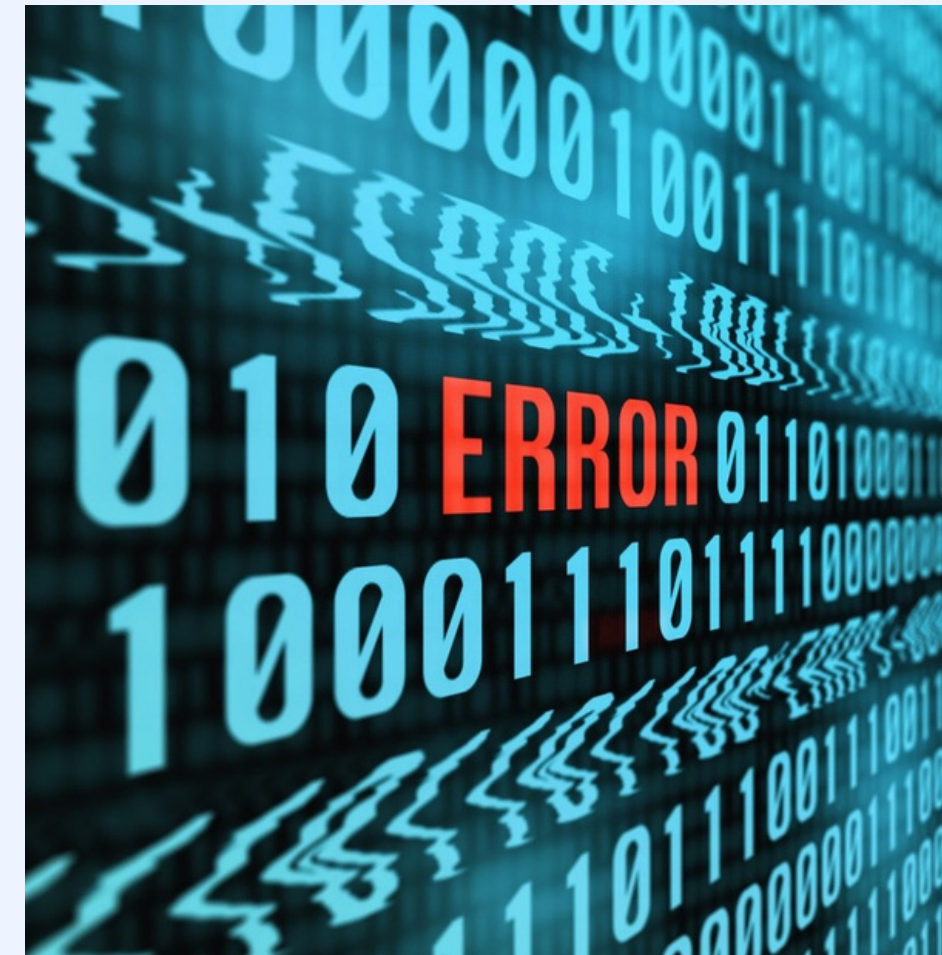
Boucle	Syntaxe	Usage typique
for	<code>for (\$i=0; \$i -lt 10; \$i++) { ... }</code>	Boucle avec compteur
foreach	<code>foreach (\$item in \$collection) { ... }</code>	Parcourir une collection
while	<code>while (condition) { ... }</code>	Boucle tant que condition vraie
do/while	<code>do { ... } while (condition)</code>	Boucle au moins une fois
do/until	<code>do { ... } until (condition)</code>	Boucle jusqu'à ce que condition vraie



Gestion des erreurs

- Utiliser try/catch/finally pour gérer les erreurs

```
try {  
    Get-Process -Name "Inexistant" -ErrorAction Stop  
}  
catch {  
    Write-Output "Erreur : $_"  
}
```





Quelques commandes
utiles

● → Commandes fichiers / dossiers

Commande	Description
Get-ChildItem	Lister fichiers/dossiers
New-Item	Créer un fichier/dossier
Remove-Item	Supprimer un fichier/dossier
Copy-Item	Copier un fichier/dossier
Move-Item	Déplacer un fichier/dossier
Rename-Item	Renommer un fichier/dossier
Get-Content	Lire le contenu d'un fichier
Set-Content	Écrire dans un fichier
Add-Content	Ajouter du contenu à un fichier

● → Commandes processus/services

Commande	Description
Get-Process	Lister les processus en cours
Stop-Process	Arrêter un processus
Get-Service	Lister les services
Start-Service	Démarrer un service
Stop-Service	Arrêter un service
Restart-Service	Redémarrer un service

● → Commandes utilisateurs

Commande	Description
New-LocalUser	Créer un utilisateur local
Remove-LocalUser	Supprimer un utilisateur local
Add-LocalGroupMember	Ajouter un utilisateur à un groupe local
Get-LocalUser	Lister les utilisateurs locaux
Get-ADUser*	Lister les utilisateurs AD
New-ADUser*	Créer un utilisateur AD
Set-ADUser*	Modifier un utilisateur AD
Remove-ADUser*	Supprimer un utilisateur AD

● → Commandes réseau/supervision

Commande	Description
Test-NetConnection	Tester la connectivité réseau
Get-NetIPConfiguration	Afficher la configuration IP
Get-NetAdapter	Lister les cartes réseau
Test-Connection	Faire un ping
Get-WmiObject	Interroger WMI (matériel, OS, etc.)
Get-Counter	Récupérer des compteurs de performance

● → Commandes archivage

Commande	Description
Compress-Archive	Compresser des fichiers/dossiers en ZIP
Expand-Archive	Décompresser un fichier ZIP
Copy-Item -Recurse	Copier récursivement un dossier

● → Commandes automatisation

Commande	Description
Start-Job	Lancer une tâche en arrière-plan
Get-Job	Lister les tâches en arrière-plan
Receive-Job	Récupérer le résultat d'une tâche
Register-ScheduledJob	Planifier une tâche