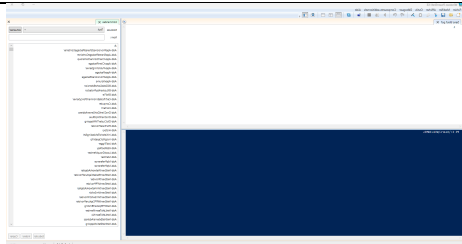
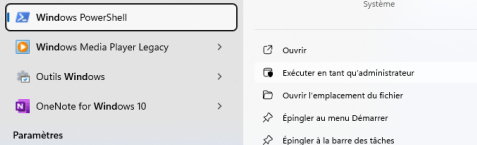
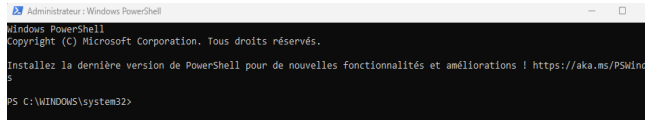
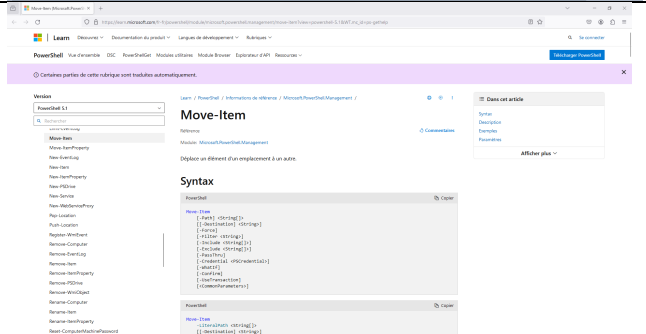
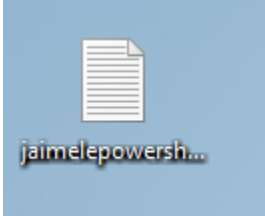


Action	Commande	Resultat
Lancer Powershell ISE	<pre>PS C:\Users\WILLIAMS> powershell_ISE.e</pre>	
Lancer PowerShell	<pre>PS C:\Users\WILLIAMS> powershell.exe</pre>	<pre>PS C:\Users\WILLIAMS></pre>
Lancer PowerShell en mode administrateur		
Lancer une commande DOS, puis PowerShell	<pre>C:\Users\WILLIAMS>powershell.exe</pre>	<pre>PS C:\Users\WILLIAMS></pre>
Lancer une commande dans PowerShell d'aide	<pre>PS C:\Users\WILLIAMS> get-help Get-ChildItem</pre>	<pre>Get-ChildItem NAME ----- DESCRIPTION Gets the items and child items in one or more specified locations. SYNTAX Get-ChildItem [[<Filter>] <System.String>] [<Attributes>] [<Archive>] [<Compressed>] [<Device>] [<Directory>] [<Encrypted>] [<Hidden>] [<IntegrityStream>] [<Normal>] [<NoClobber>] [<NoContentIndexed>] [<Offline>] [<ReadOnly>] [<ReparsePoint>] [<SparseFile>] [<System>] [<Temporary>] [<Depth>] [<System.ItemType>] [<Directory>] [<Exclude>] [<System.String>] [<File>] [<Force>] [<Hidden>] [<Include>] <System.String>] [<LiteralPath>] [<System.String>] [<Name>] [<ReadOnly>] [<Recurse>] [<System>] [<UseTransaction>] [<CommonParameters>] Get-ChildItem [-Path] <System.String>] [[<Filter>] <System.String>] [<Attributes>] [<Archive>] [<Compressed>] [<Device>] [<Directory>] [<Encrypted>] [<Hidden>] [<IntegrityStream>] [<Normal>] [<NoClobber>] [<NoContentIndexed>] [<Offline>] [<ReadOnly>] [<ReparsePoint>] [<SparseFile>] [<System>] [<Temporary>] [<Depth>] [<System.ItemType>] [<Directory>] [<Exclude>] [<System.String>] [<File>] [<Force>] [<Hidden>] [<Include>] [<System.String>] [<Name>] [<ReadOnly>] [<Recurse>] [<System>] [<UseTransaction>] [<CommonParameters>] DESCRIPTION The Get-ChildItem cmdlet gets the items in one or more specified locations. If the item is a container, it gets the items inside the container, known as child items. You can use the Recurse parameter to get items in all child containers and use the Depth parameter to limit the number of levels to recurse. PS C:\Users\WILLIAMS> get-help Move-Item</pre>
Lancer l'aide de la commande « Move-item »	<pre>PS C:\Users\WILLIAMS> get-help Move-Item</pre>	<pre>Move-Item NAME ----- DESCRIPTION Moves an item from one location to another. SYNTAX Move-Item [[<Destination>] <System.String>] [<Credential>] <System.Management.Automation.PSCredential>] [<Exclude>] <System.String>] [<Filter>] <System.String>] [<Force>] [<Include>] <System.String>] [<LiteralPath>] <System.String>] [<Confirm>] [<ConfirmAction>] [<Confirm>] [<WhatIf>] [<CommonParameters>] Move-Item [-Path] <System.String>] [[<Destination>] <System.String>] [<Credential>] <System.Management.Automation.PSCredential>] [<Exclude>] <System.String>] [<Filter>] <System.String>] [<Force>] [<Include>] <System.String>] [<File>] [<Force>] [<UseTransaction>] [<Confirm>] [<WhatIf>] [<CommonParameters>] DESCRIPTION The Move-Item cmdlet moves an item, including its properties, contents, and child items, from one location to another location. The locations must be supported by the same provider. For example, it can move a file or subdirectory from one directory to another or move a registry subkey from one key to another. When you move an item, it is added to the new location.</pre>
Ouvrez l'aide de la commande « Move-Item » sur internet	<pre>PS C:\Users\WILLIAMS> get-help Move-Item -Online</pre>	
Lister le contenu d'un dossier	<pre>PS C:\Users\WILLIAMS> Get-ChildItem "c:"</pre>	<pre>Répertoire : C:\Users\WILLIAMS Mode LastWriteTime Length Name ---- - d----- 05/09/2023 13:12 .ms-ad d----- 05/12/2023 16:26 .VirtualBox d-r----- 04/09/2023 09:13 Contacts d-r----- 05/12/2023 09:41 Desktop d-r----- 22/01/2024 14:03 Documents d-r----- 04/03/2024 15:16 Downloads d-r----- 04/09/2023 09:13 Favorites d-r----- 04/09/2023 09:13 Links d-r----- 04/09/2023 09:13 Music d-r----- 05/12/2023 09:31 OneDrive dar--l 04/03/2024 15:46 OneDrive - Les Charmilles d-r----- 20/11/2023 10:10 Pictures d-r----- 04/09/2023 09:13 Saved Games d-r----- 04/09/2023 09:19 Searches d-r----- 26/09/2023 14:07 Videos d----- 26/09/2023 13:37 VirtualBox VMs</pre>

Action	Commande Powershell	Résultat
Lister les fichiers présent dans le dossier c: \Temp	<code>PS C:\Users\WILLIAMS> Get-ChildItem "C:\Windows\Temp"</code>	<pre> Répertoire : C:\Windows\Temp Mode LastWriteTime Length Name ---- - d----- 04/03/2024 11:04 temp00000075 d----- 04/03/2024 11:00 vmtoolsd-SetTime d----- 04/03/2024 11:12 8578 AdobePDF_1og d----- 04/03/2024 11:15 2088 Application_Mylog_1og d----- 04/03/2024 11:15 21041112 Application_SIX W884-6E2C-0000-11E0-C50C-6E0001.exe d----- 04/03/2024 11:05 21041112 Application_SIX W884-6E2C-0000-11E0-C50C-6E0001.exe d----- 04/03/2024 11:05 842 ApplicationReport_SIX W884-6E2C-0000-11E0-C50C-6E0001.txt d----- 04/03/2024 11:05 842 ApplicationReport_SIX W884-6E2C-0000-11E0-C50C-6E0001.txt d----- 04/03/2024 11:08 240294 Armit_1st d----- 04/03/2024 11:12 1248 ClientServerReport_1st d----- 04/03/2024 11:12 208078 DEKTOP-7URAQR5-20240304-1306_1og d----- 04/03/2024 11:12 15644 DEKTOP-7URAQR5-20240304-1312_1og d----- 04/03/2024 11:11 16078 DEKTOP-7URAQR5-20240304-1311_1og d----- 04/03/2024 11:12 16002 DEKTOP-7URAQR5-20240304-1312a_1og d----- 04/03/2024 11:12 15803 DEKTOP-7URAQR5-20240304-1312b_1og d----- 04/03/2024 11:16 15851 DEKTOP-7URAQR5-20240304-1320_1og d----- 04/03/2024 11:16 15812 DEKTOP-7URAQR5-20240304-1318_1og d----- 04/03/2024 14:10 17292 DEKTOP-7URAQR5-20240304-1444_1og d----- 04/03/2024 11:00 12292 DEKTOP-7URAQR5-20240304-1100_1og d----- 17/04/2023 11:12 0 F327777DeDup.opf1a_1st d----- 25/04/2023 11:16 0 F327777DeDup.opf1a_1st d----- 04/03/2024 11:05 1112118 Reporte de la mise à jour de Windows, Admin, SIX W884-6E2C-0000-11E0-C50C-6E0001.exe </pre>
Demander le nom de votre PC	<code>PS C:\Users\WILLIAMS> hostname</code>	DESKTOP-7URAQR5
Redémarrer votre PC au bout de 30 secondes		
Crée un dossier dans C:\ Nommer « MesScripts »	<code>New-Item c:\ -Name mesScript</code>	<pre> Répertoire : C:\ Mode LastWriteTime Length Name ---- - -a----- 04/03/2024 15:55 0 mesScript </pre>
Demander la date année mois jours heures secondes et fuseau horaire	<code>PS C:\WINDOWS\system32> Get-Date</code>	lundi 4 mars 2024 15:58:38
Crée un fichier sur votre bureau nommer « j'aime le powershell.txt »	<code>New-Item "C:\Users\WILLIAMS\Desktop" -Name jaimelle powershell.txt</code>	
Afficher la Build Windows sur le bureau windows (modification par la base de registre)	<code>[System.Environment]::OSVersion.Version</code>	<pre> Major Minor Build Revision ----- 10 0 19045 0 </pre>
Ajouter la fonctionnalité windows service d'Impression en PDF	<code>Enable-WindowsOptionalFeature -Online -FeatureName PrintingToPDFServices-Features</code>	<pre> Path : Online : True RestartNeeded : False </pre>
Récupérer votre adresse IP v4	<code>Get-NetIPAddress -AddressFamily IPV4</code>	<pre> IPAddress : 10.1.42.109 InterfaceIndex : 19 InterfaceAlias : Ethernet 2 AddressFamily : IPv4 Type : Unicast PrefixLength : 23 PrefixOrigin : Dhcp SuffixOrigin : Dhcp AddressState : Preferred ValidLifetime : 4.21:16:35 PreferredLifetime : 4.21:16:35 SkipAsSource : False PolicyStore : ActiveStore </pre>

Manipulation de fichiers / dossiers		
Apprendre	Commande en PS	Remarque
Crée un dossier	<code>New-Item "C:\Users\jack.williams\Desktop\TEST" -itemType Directory</code>	J'ai créer un dossier sur le Bureau
Crée un fichier	<code>New-Item -Path "C:\Users\jack.williams\Desktop\TEST\test.txt" -ItemType File</code>	J'ai créer un fichier .txt dans le dossier TEST
Bouger un fichier	<code>Get-Item -Path C:\Users\jack.williams\Desktop\TEST\test.txt Move-Item -Destination C:\Users\jack.williams\Documents</code>	J'ai déplacer le fichier test.txt dans le dossier Documents
Bouger un dossier	<code>Get-Item -Path C:\Users\jack.williams\Desktop\TEST Move-Item -Destination C:\Users\jack.williams\Documents</code>	J'ai déplacer le dossier TEST dans le dossier Documents
Renommer un fichier	<code>Rename-Item -Path C:\Users\jack.williams\Documents\test.txt Powershell</code>	J'ai renommer le fichier test.txt en powershell.txt
Renommer un dossier	<code>Rename-Item -Path C:\Users\jack.williams\Documents\TEST Powershell_file</code>	J'ai renommer le dossier TEST en Powershell_file
Supprimer un fichier	<code>Remove-Item -Path C:\Users\jack.williams\Documents\Powershell</code>	J'ai supprimer le fichier Powershell
Supprimer un dossier	<code>Remove-Item -Path C:\Users\jack.williams\Documents\Powershell_file</code>	J'ai supprimer le dossier Powershell_file

TP POWERSHELL 2

Demande	Comman de en PS	Résultat
Les commenta ires	#Test	<pre>PS C:\WINDOWS\system32> #Test</pre>
Les Variables	\$	<pre>1 #Cette commande nous créer une variable qui contient "Hello". script lancer par Jack 2 \$mavariab1e1 = "Hello"</pre>
Get- Module	Get- module	<pre>#Cette commande nous montre tout les fichier dans le repertoire C:\Program Files\WindowsPowerShell\Modules et C:\WINDOWS\system32\windowsPowerShell\v1.0\Modules. PS C:\WINDOWS\system32> #Cette commande nous montre tout les fichier dans le repertoire C:\Program Files\WindowsPowerShell\Modules et C:\WINDOWS\sy Get-Module -ListAvailable Répertoire : C:\Program Files\WindowsPowerShell\Modules ----- ModuleType Version Name ExportedCommands ----- Script 1.0.1 Microsoft.PowerShell.Operation.V... {Get-OperationValidation, Invoke-OperationValidation} Binary 1.0.0.1 PackageManagement {Find-Package, Get-Package, Get-PackageProvider, Get-PackageSource...} Script 3.4.0 Pester {Describe, Context, It, Should...} Script 1.0.0.1 PowerShellGet {Install-Module, Find-Module, Save-Module, Update-Module...} Script 2.0.0 PSReadLine {Get-PSReadLineKeyHandler, Set-PSReadLineKeyHandler, Remove-PSReadLineKeyHandler, Ge... Répertoire : C:\WINDOWS\system32\WindowsPowerShell\v1.0\Modules ----- ModuleType Version Name ExportedCommands ----- Manifest 1.0.0.0 AppBackgroundTask {Disable-AppBackgroundTaskDiagnosticLog, Enable-AppBackgroundTaskDiagnosticLog, Set-... Manifest 2.0.0.0 AppLocker {Get-AppLockerFileInformation, Get-AppLockerPolicy, New-AppLockerPolicy, Set-AppLock... Manifest 1.0.0.0 AppxClient {Add-AppClientConnectionGroup, Add-AppClientPackage, Add-AppPublishingServer, Dis... Manifest 2.0.1.0 Appx {Add-AppPackage, Get-AppPackage, Get-AppPackageAutoUpdateSettings, Get-AppPackage...</pre>
Clear-host	Clear- host	<pre>#Cette commande nous supprime tout les commande au paravant . Script lancer par Jack Clear-Host</pre>
Write-host vs Echo	Echo et write- host	<pre>#Cette commande nous affiche ce qu'on lui demande. Script lancer par Jack write-host test echo Test 2</pre>
Read-host	Read- host	<pre>#Cette commande nous créer un prompt qui nous demande d'entrer une info. Read-host "Entrez votre age."</pre>
Les Pipes		<pre>PS C:\WINDOWS\system32> #Cette commande nous créer un prompt qui nous demande d'entrer une info. Script lancer par Jack Get-AppPackage Select Name, PackageFullName Name PackageFullName ---- Microsoft.Windows.008NetworkConnectionFlow Microsoft.Windows.008NetworkConnectionFlow_10.0.21302.1000_neutral__cw5n1h2txyewy Microsoft.Windows.008NetworkCaptivePortal Microsoft.Windows.008NetworkCaptivePortal_10.0.21302.1000_neutral__cw5n1h2txyewy Microsoft.Windows.UndockedDevKit Microsoft.Windows.UndockedDevKit_10.0.22621.1_neutral_neutral__cw5n1h2txyewy Microsoft.UI.Xaml.CBS Microsoft.UI.Xaml.CBS_8.2303.31001.0_x64__8wekyb3d8bbwe Microsoft.WindowsAppRuntime.CBS Microsoft.WindowsAppRuntime.CBS_4000.864.1406.0_x64__8wekyb3d8bbwe Microsoft.Windows.Client.FileExp Microsoft.Windows.Client.FileExp_1000.22653.1000.0_x64__cw5n1h2txyewy Microsoft.Wallet Microsoft.Wallet_2.4.18324.0_x64__8wekyb3d8bbwe Microsoft.BioEnrollment Microsoft.BioEnrollment_10.0.19587.1000_neutral__cw5n1h2txyewy</pre>
Format- List	get-VM fl *	<pre>PS C:\WINDOWS\system32> #Cette commande nous affiche tout les informations sur les VM qui sont stocker sur le poste. Script lancer par Jack Get-VM fl * ParentCheckpointId : ac0e67eb-a411-4014-9dee-2e1aa33815f6 ParentCheckpointName : Point de contrôle automatique - A00501 - (19/02/2024 - 13:32:10) VMName : A00501 VMId : 728d34ef-9335-4c89-ab3a-245cc6a066bc CheckpointFileLocation : C:\ProgramData\Microsoft\Windows\Hyper-V ConfigurationLocation : C:\ProgramData\Microsoft\Windows\Hyper-V GuestStatePath : C:\ProgramData\Microsoft\Windows\Hyper-V\Virtual Machines\728D34EF-9335-4C89-AB3A-245CC6A066BC\vmgs SmartPagingFileInUse : False SmartPagingFilePath : C:\ProgramData\Microsoft\Windows\Hyper-V SnapshotFileLocation : C:\ProgramData\Microsoft\Windows\Hyper-V AutomaticStartAction : StartIfRunning AutomaticStartDelay : 0 AutomaticStopAction : Save AutomaticCriticalErrorAction : Pause AutomaticCriticalErrorActionTimeout : 30 AutomaticCheckpointsEnabled : True</pre>
Opérateur s de comparais on		<pre>-eq = equal -ne-gt = not equal-greater than -lt-le = less than-lower-equal -ge = greater equal -like= comme -notlike = pas comme -in= dans -notin= pas dedans - contains = contient -notcontains = ne contient pas -is = est -isnot = n'est pas</pre>

Where-Object	Get-Service Where-object	<pre>PS C:\WINDOWS\system32> #Cette commande nous affiche tout les processus qui tourne en arrière plan. Script lancer par Jack Get-Service Where-object -Property status -EQ 'Running' Status Name DisplayName ----- Running AdobeARMService Adobe Acrobat Update Service Running agent_ovpnconnect OpenVPN Agent agent_ovpnconnect Running Appinfo Informations d'application Running AppXSvc Service de déploiement AppX (AppXSVC) Running AudioEndpointBu... Générateur de points de terminaison... Running Audiosrv Audio Windows Running BDESVC Service de chiffrement de lecteur B... Running BFE Moteur de filtrage de base Running BluetoothUserSe... BluetoothUserService_4f33a Running BluetoothUserSe... Service de support des utilisateurs... Running BrEndpointSvc HP Sure Click Endpoint Service Running BrokerInfrastru... Service d'infrastructure des tâches... Running Browser Explorateur d'ordinateurs</pre>
Opérateurs de redirection	>>	<pre>PS C:\WINDOWS\system32> #Cette commande nous amène tout les éléments dans fichier.txt. Script lancer par Jack Get-process >> .\Services.txt</pre>
Enumeration	Enum	
Format-Table	 Format-table	<pre>PS C:\WINDOWS\system32> #Cette commande nous affiche les VM en table. Script lancer par Jack Get-VM Format-Table -AutoSize Name State CPUUsage(%) MemoryAssigned(M) Uptime Status Version ----- ADD501 Running 0 3386 02:50:20.3180000 Fonctionnement normal 11.0 ADRODC Running 0 3124 02:50:19.7810000 Fonctionnement normal 11.0 SVRFILES Running 0 2246 02:50:19.6760000 Fonctionnement normal 11.0</pre>

TP powershell 3

While :

```
1 $i = 0
2 $j = 0..99
3 while($i -lt $j.Length){
4     $j[$i]
5     $i++
6 }
7
8 Write-Host $i
```

Ce script ajoute +1 entre le 0 et 99 avec la boucle while

Do While :

```
Clear-Host

$i = 0
$j = Read-Host "Entrez un nombre"

Write-Host "Vous avez entrer le nombre $j le compte se fait de $i a $j tant que $i est plus pe'

do {
    "La valeur est : $i/$j"
    $i++
} while ($i -le $j)

Write-Host "Fin de la boucle :)"
```

```
Entrez un nombre: 12
Vous avez entrer le nombre 12 le compte se fait de 0 a 12 tant que 0 est plus petit ou egale a 12
La valeur est : 0/12
La valeur est : 1/12
La valeur est : 2/12
La valeur est : 3/12
La valeur est : 4/12
La valeur est : 5/12
La valeur est : 6/12
La valeur est : 7/12
La valeur est : 8/12
La valeur est : 9/12
La valeur est : 10/12
La valeur est : 11/12
La valeur est : 12/12
Fin de la boucle :)
```

Cette commande ajoute +1 jusqu'au nombre choisi. Avec la commande do while.

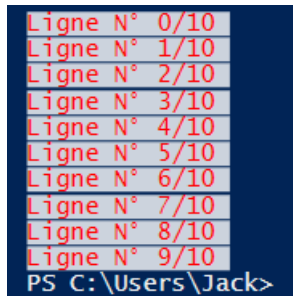
Do Until :

```
Entrez un nombre pour faire le décompte: 12
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
i = 11
i = 12
La boucle est arrivée à 12, le décompte est terminé !
PS C:\Users\Jack>
```

Ce script ajoute +1 jusqu'aux nombre qu'on a choisi.

For

```
clear For ($i=0; $i -lt 10; $i++)  
{ Write-host "Ligne N° $i/10 " -ForegroundColor red -BackgroundColor white }
```



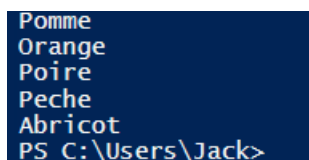
```
Ligne N° 0/10  
Ligne N° 1/10  
Ligne N° 2/10  
Ligne N° 3/10  
Ligne N° 4/10  
Ligne N° 5/10  
Ligne N° 6/10  
Ligne N° 7/10  
Ligne N° 8/10  
Ligne N° 9/10  
PS C:\Users\Jack>
```

Ce script nous choisi 10 ligne et surligne les ligne choisi.

For Each

```
Clear  
$fruits= "Pomme", "Orange", "Poire", "Pêche", "Abricot"  
foreach ($fruit in $fruits)
```

```
{  
echo $fruit  
}
```



```
Pomme  
Orange  
Poire  
Pêche  
Abricot  
PS C:\Users\Jack>
```

Ce script nous imprime chaque fruit qui est dans la liste \$fruit

Exemple if, Elseif, Then

```
Clear-Host  
$fruit = Read-Host "Entrez le nom d'un fruit"
```

```
if ($fruit -eq "Banane") {  
    Write-Host "C'est très bon la banane !"  
}  
elseif ($fruit -eq "Poire") {  
    Write-Host "C'est encore meilleur une poire !"  
}  
elseif ($fruit -eq "Kiwi") {
```

```

    Write-Host "C'est bon le kiwi !"
}
elseif ($fruit -eq "Pêche") {
    Write-Host "C'est bon la pêche !"
}
elseif ($fruit -eq "Abricot") {
    Write-Host "C'est bon l'abricot !"
}
else {
    Write-Host "Ce fruit $fruit n'est pas dans la liste !"
}

```

```

Entrez le nom d'un fruit: banane
C'est très bon la banane !

PS C:\Users\Jack>

```

Ce script nous demande de choisir un fruit dans la liste et répond c'est très bon et si le fruit n'est pas dans la liste il répond le fruit n'est pas dans la liste.

Switch

```

$nombre = Read-Host -Prompt "Entrez un nombre entre 1 et 5"
switch ($nombre) {
    1 { Write-Host "Bravo, Vous avez entré le Nombre 1" ; Break }
    3 { Write-Host "Bravo, Vous avez entré le Nombre 3" ; Break }
    4 { Write-Host "Bravo, Vous avez entré le Nombre 4" ; Break }
    5 { Write-Host "Bravo, Vous avez entré le Nombre 5" ; Break }
    Default { Write-Host "Vous avez entré un nombre incorrect : $nombre, qui n'est pas entre 1 et 5 !" }
}
Entrez un nombre entre 1 et 5: 4
Bravo, Vous avez entré le Nombre 4

PS C:\Users\Jack>

```

Ce script nous demande de choisir un nombre entre 1 et 5 ensuite il défile la liste donnée jusqu'à qu'il trouve notre nombre entrée

Fonctions :

```

PS C:\Users\Jack>
function Bonjour {
    $date = Get-Date
    Write-Host "Bonjour, nous sommes le $date"
}

Bonjour
Bonjour, nous sommes le 04/08/2024 14:22:21

PS C:\Users\Jack>

```

Ce script récupère la date d'aujourd'hui et nous l'imprime lorsqu'il est appelé.

Les fonctions avec Arguments Dans Powershell :

```

Bonjour Jack Williams, nous sommes le 04/08/2024 14:35:00
PS C:\Users\Jack>

Clear

function Bonjour
{
    $date = Get-Date
    Write-Host "Bonjour $args, nous sommes le $date"
}

Bonjour "Jack Williams"

```

Ce script est exactement comme le fonction des dates mais nous ajoute un argument « \$arg » pour ajouter notre nom.

Exemple des paramètres:

```
Titre du message: ceci est un titre de message
Un message pour vous : Bonjour a tous, ceci est un contenu de message !
PS C:\Users\Jack>

Clear

function Show-Popup
{
Param ($message, $titre)
Write-Host "Titre du message:" $titre
Write-Host "Un message pour vous :" $message
}

Show-Popup -message "Bonjour a tous, ceci est un contenu de message !" -titre "ceci est un titre de m
```

Ce script nous imprime des titre avec des fonctions appeler \$titre

Script que j'ai créer qui te donne l'option d'ouvrir des applications Office 365 :

```
1 function Open-OfficeApp {
2     param (
3         [Parameter(Mandatory = $true)]
4         [string]$appName
5     )
6
7     switch ($appName) {
8         "Excel" { Start-Process "Excel" }
9         "Access" { Start-Process "msaccess" }
10        "Publisher" { Start-Process "mspub" }
11        "OneNote" { Start-Process "onenote" }
12        "Outlook" { Start-Process "outlook" }
13        "PowerPoint" { Start-Process "powerpnt" }
14        "Word" { Start-Process "winword" }
15        default { Write-Host "Application pas reconnu." }
16    }
17 }
18
19 $AppToOpen = Read-Host "Quelle application Office 365 voulais vous ouvrir? (Excel, Access, Publ
20
21 Open-OfficeApp -appName $AppToOpen
```