

Referencia Rápida de PowerShell

Cmdlets, pipeline, objetos, scripting, módulos

Fundamentos

Comandos y Ayuda

```
Get-Help Get-Process # show help for cmdlet
Get-Help Get-Process -Online # open online docs
Get-Command *service* # find commands by name
Get-Alias ls # show alias target
```

Alias Comunes

```
ls → Get-ChildItem Listar archivos y directorios
cd → Set-Location Cambiar directorio
cp → Copy-Item Copiar archivo o directorio
mv → Move-Item Mover o renombrar
rm → Remove-Item Eliminar archivo o directorio
cat → Get-Content Leer contenido del archivo
echo → Write-Output Imprimir al pipeline
cls → Clear-Host Limpiar la consola
```

Variables

Básicos de Variables

```
$name = "Alice" # string
$count = 42 # integer
$pi = 3.14 # double
$list = @(1, 2, 3) # array
$hash = @{a=1; b=2} # hashtable
```

Variables Automáticas

```
$_ Objeto actual del pipeline
$PSVersionTable Información de versión de PowerShell
$HOME Directorio de inicio del usuario
$PWD Directorio actual
$null Valor nulo
$true / $false Constantes booleanas
$error Array de errores recientes
$LASTEXITCODE Código de salida del último comando nativo
```

Variables de Entorno

```
$env:PATH # read env var
$env:MY_VAR = "value" # set env var
Get-ChildItem Env: # list all env vars
```

Operadores

Operadores de Comparación

```
-eq -ne Igual / no igual
-gt -lt Mayor / menor que
-ge -le Mayor/menor o igual
-like -notlike Coincidencia con comodín (*, ?)
-match -notmatch Coincidencia con regex
-contains La colección contiene el valor
-in -notin Valor en la colección
```

Operadores Lógicos y Otros

```
-and -or -not Operadores lógicos
! NOT lógico (alias)
-replace Reemplazo con regex: 'hi' -replace 'h','b'
-split -join Dividir / unir cadenas
.. Rango: 1..5 → 1,2,3,4,5
?: Ternario (v7+): $x ? 'yes' : 'no'
```

Flujo de Control

if / elseif / else

```
if ($age -ge 18) {
    "Adult"
} elseif ($age -ge 13) {
    "Teen"
} else {
    "Child"
}
```

switch

```
switch ($color) {
    "red" { "Stop" }
    "green" { "Go" }
    default { "Unknown" }
}
```

Bucles

```
foreach ($item in $list) { $item }
for ($i=0; $i -lt 5; $i++) { $i }
while ($x -lt 10) { $x++ }
1..5 | ForEach-Object { $_ * 2 }
```

Funciones

Definir Funciones

```
function Get-Greeting {
    param([string]$Name = "World")
    "Hello, $Name!"
}
Get-Greeting -Name "Alice"
```

Parámetros Avanzados

```
function Copy-SafeFile {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory)][string]$Path,
        [Parameter(Mandatory)][string]$Dest
    )
    Copy-Item $Path $Dest -WhatIf:$WhatIfPreference
}
```

Objetos y Pipeline

Básicos del Pipeline

```
Get-Process | Sort-Object CPU -Desc | Select-Object -First 5
Get-Service | Where-Object Status -eq "Running"
Get-ChildItem | Measure-Object -Property Length -Sum
```

Cmdlets Clave del Pipeline

```
Where-Object Filtrar objetos: | Where {$_.CPU -gt 10}
Select-Object Elegir propiedades: | Select Name, CPU
Sort-Object Ordenar: | Sort CPU -Desc
ForEach-Object Transformar cada uno: | ForEach {$_.Name}
Measure-Object Contar, sumar, promedio, mín, máx
Group-Object Agrupar por valor de propiedad
Format-Table Mostrar como tabla: | ft -Auto
Export-Csv Exportar a CSV: | Export-Csv out.csv
```

Archivos

Operaciones con Archivos

```
Get-Content log.txt # read file
Set-Content out.txt "hello" # write (overwrite)
Add-Content out.txt "more" # append
Get-Content log.txt | Select-String "error" # grep
```

Cmdlets de Ruta y Archivo

```
Test-Path $path Verificar si el archivo/dir existe
New-Item -Type File Crear archivo
New-Item -Type Directory Crear directorio
Resolve-Path Obtener ruta absoluta
Join-Path Combinar segmentos de ruta
Split-Path Obtener padre o hoja
Get-ItemProperty Atributos y metadatos del archivo
Remove-Item -Recurse Eliminar directorio recursivamente
```

Cadenas

Básicos de Cadenas

```
"Hello, $name" # interpolation (double quotes)
'Hello, $name' # literal (single quotes)
"Length: $($list.Count)" # expression in string
@"
Multi-line
here-string with $name
"@
```

Métodos de Cadenas

```
.ToUpper() / .ToLower() Cambiar mayúsculas/minúsculas
.Trim() Eliminar espacios al inicio y final
.Split(',') Dividir en array
.Replace('a','b') Reemplazar subcadena
.StartsWith() / .EndsWith() Verificar prefijo / sufijo
.Substring(0,5) Extraer subcadena
.Contains('text') Verificar si contiene
-f operator '{0} is {1}' -f 'sky', 'blue'
```

Módulos

Gestión de Módulos

```
Get-Module -ListAvailable # installed modules
Find-Module -Name Az* # search gallery
Install-Module -Name Pester # install from gallery
Import-Module ActiveDirectory # load module
```

Comandos de Módulos

```
Get-Module Listar módulos cargados
Import-Module Cargar módulo en la sesión
Remove-Module Descargar módulo de la sesión
Update-Module Actualizar módulo instalado
Get-Command -Module X Listar comandos del módulo
$env:PSModulePath Rutas de búsqueda de módulos
```

Patrones Comunes

Manejo de Errores

```
try {
    Get-Item "C:\missing" -ErrorAction Stop
} catch {
    "Error: $_"
} finally {
    "Cleanup here"
}
```

Referencia Rápida de PowerShell

Política de Ejecución y Remoting

Get-ExecutionPolicy	Mostrar política de scripts actual
Set-ExecutionPolicy RemoteSigned	Permitir scripts locales
Enter-PSSession -Computer SRV1	Sesión remota interactiva
Invoke-Command -Computer SRV1 -Script {}	Ejecutar bloque de script remotamente
Start-Job { long-task }	Ejecutar trabajo en segundo plano
Receive-Job -Id 1	Obtener salida del trabajo en segundo plano