

Creación de scripts en PowerShell

Manuel Domínguez

 @mafradoti

<https://github.com/mftienda>

BIENVENIDOS!

Soy Manuel Domínguez.

En esta clase, vamos a explicar: **Las operaciones con variables.**

ÍNDICE

1.- Operadores

2.- Operadores aritméticos.

3.- Operadores de asignación.

4.- Operadores de comparación.

5.- Operadores lógicos.

6.- Operadores de tipos.

4.- Operadores de comparación.

Nos permiten comparar objetos y propiedades.

Nos devolverá **true o false**, según el caso.

PowerShell utiliza **los mismos operadores para comparar números que cadenas de caracteres.**

PowerShell no distingue entre mayúsculas o minúsculas.
Si nos interesa la distinción, debemos indicársela .

4.- Operadores de comparación.

Operador	Descripción	Ejemplo
-eq	Igual	5 -eq 5 "heidi" -eq "heidi"
-ne	Distinto	5 -ne 4
-gt	Mayor	5 -gt 5
-ge	Mayor o igual	5 -ge 5
-lt	Menor	5 -lt 5
-le	Menor o igual	5 -le 5

4.- Operadores de comparación.

Reto:

Buscar qué operador nos permite comparar dos cadenas, diferenciando entre mayúsculas y minúsculas.

“Heidi” -operador “HEIDI” → Debe devolver un False

Solución reto: Al final del capítulo.

Solución al reto.

Reto: Operador para distinguir mayúsculas de minúsculas.

"heidi" -eq "HEIDI" \rightarrow True

"heidi" -ceq "HEIDI" \rightarrow False

5.- Operadores lógicos.

Nos permiten comparar varias expresiones.

Operador	Descripción	Ejemplo
-and	Devuelve True si todas las expresiones son ciertas.	(7 -eq 7) -and ("h" -eq "h")
-or	Devuelve True si alguna de las expresiones es cierta.	(7 -eq 7) -or ("hola" -eq "adios")
-not !	Devuelve True si la expresión es falsa.	!(7 -eq 9)
-xor	Devuelve True cuando sólo una de las expresiones es falsa.	(7 -eq 7) -xor (4 -eq 4) → False (7 -eq 7) -xor (4 -eq 3) → True

6.- Operadores de tipos.

Nos sirve para comprobar el tipo de una variable.

Operador	Descripción	Ejemplo
-is	Devuelve True si la variable es de un determinado tipo.	5 -is [int] → True '5' -is [int] → False
-isNot	Devuelve True si la variable no es del tipo que se indica.	"hola" -isnot [int] → True

Práctica: Operadores

Realizar un script: **ArchivosGrandes.ps1**

Entrada: El script preguntará:

Ruta para buscar archivos.

Tamaño a partir del cual buscaremos: 100MB,2GB.

Salida: Muestra aquellos archivos que sean mayores que el indicado.

Se exportará los resultados a un archivo html.

Pista: **Where-object** y **ConvertTo-HTML**

Práctica: Operadores

ArchivosGrandes.ps1 X

```
7 #Definición de funciones
8 #Bloque principal
9 Clear
10 Write-Host "Búsqueda de archivos grandes"
11 #Toma de datos
12 $ruta=Read-Host "Introduce la ruta absoluta de búsqueda"
13 $tamano=Read-Host "Introduce el tamaño a partir del cuál buscaremos (Ejemplos: 100MB,1GB)"
14
15 #Buscamos los archivos y lo ordenamos
16 #ls -r $ruta |Where-object {$_.Length -gt $tamano}|sort-object -Property Length|ft name,length
17 #Creamos un archivo html con los resultados
18 ls -r $ruta |Where-object {$_.Length -gt $tamano}|sort-object -Property Length | #Provoca una nueva línea automáticamente
19 ConvertTo-Html -Property name,length -body "<H2> Archivos grandes </H2>"|
20 out-file ArchivosGrandes.html; Invoke-Item ArchivosGrandes.html
```

Resumen

Operadores aritméticos: +, -, *, /, %

Operadores de la clase [math]: floor, ceiling, sqrt, pow, round, etc.

Operadores de asignación: =, +=, -=, *=, /=

Operadores de comparación: -eq, -ne, -gt, -ge, -lt, -le

Operadores lógicos: -and, -or, -not (!), -xor

Operadores de tipo: -is, -isNot

Creación de scripts en PowerShell

Manuel Domínguez

 @mafradoti

<https://github.com/mftienda>

DESPEDIDA!

Hemos llegado al final de este vídeo.
Nos vemos en el siguiente.