

Creación de scripts en PowerShell

Manuel Domínguez

 @mafradoti

<https://github.com/mftienda>

BIENVENIDOS!

Soy Manuel Domínguez.

En esta clase, vamos a explicar: **La creación y asignación de variables.**

ÍNDICE

- 1.- Creación y asignación de variables.
- 2.- Contenido de una variable.
- 3.- Determinar el tipo de una variable.
- 4.- Cmdlets para trabajar con variables.
- 5.- Lectura y escritura de variables en un script.

Resumen

1.- Creación y asignación de variables.

Una variable es un espacio de la memoria principal a la que le asignamos un nombre para poder acceder.

Empiezan por el símbolo \$

No es obligatorio declararla.

PowerShell se encarga de asignarle el tipo.

Recomendación: El nombre de la variable lo escribiremos en minúsculas.

Ejemplos:

`$edad=52`

`$nombre="Manuel Domínguez"`

`${mi nombre}= "Manuel Domínguez"` → Si utilizamos espacios en blanco, el nombre de la variable debe ir entre llaves.

1.- Creación y asignación de variables: constantes.

Una **constante** es una variable que no cambia de valor. Se trata de una variable de solo lectura.

Recomendación: El nombre de la constante lo escribiremos en mayúsculas.

Utilizaremos el cmdlet: **New-Variable**, con la Option Constant

Ejemplo:

New-Variable -Name DESTINO -Value "d:\Seguridad\copias\" -

Option Constant

\$DESTINO

\$DESTINO.GetType()

2.-Contenido de una variable.

Para ver el contenido de una variable, es suficiente con escribir el nombre de la variable con el \$ delante.

Ejemplos:

\$edad

\$nombre

\${mi nombre}

\$DESTINO

3.- Determinar el tipo de una variable.

Como hemos dicho, PowerShell le asigna un tipo a la variable.

Para conocer el tipo asignado, utilizaremos el método: **GetType()**

Ejemplos:

`$edad.GetType()` → Nos fijamos en la columna **Name**.

`$nombre.GetType()`

`${mi nombre}.GetType()`

`$DATOS.GetType()`

Práctica: Variables y constantes.

Cambiamos el valor de la variable edad:

```
$edad=53  
$edad
```

Intentamos cambiar el valor de la constante DESTINO:

```
$DESTINO="d:\copias"
```

No nos deja.

4.- Cmdlets para trabajar con variables.

Get-Command *variable

Clear-Variable: Limpia el contenido de una variable.

Get-Variable: Nos muestra las variables (Automáticas de PowerShell + Creadas por el usuario) en la consola actual.

New-Variable: Nos permite crear explícitamente una variable.

Remove-Variable: Elimina la variable de la consola actual.

Set-Variable: Modifica o establece las propiedades de una variable.

6.- Lectura y escritura de una variable en un script.

Read-Host: Lee una línea de entrada de la consola. Se puede utilizar para preguntar al usuario un dato. Se guarda en una variable como **texto**.

-AsscureString: Con esta opción nos aseguramos que la cadena introducida se guarde de forma segura.

Write-Host: Podemos utilizarlo para que nos muestre un texto en pantalla o bien el contenido de una variable.

``n`` → Salto de línea

``t`` → Tabulador

6.- Lectura y escritura de una variable en un script.

Write-Host y las comillas simples ' y las comillas dobles "

```
PS C:\Users\Windows> $x=10
PS C:\Users\Windows> Write-Host "El valor de x es: $x"
El valor de x es: 10
PS C:\Users\Windows> Write-Host 'El valor de x es: $x'
El valor de x es: $x
```

Write-Host y acceso a una propiedad dentro de las "

```
PS C:\Users\Windows> $archivo=ls .\prueba.txt
PS C:\Users\Windows> $archivo.name
prueba.txt
PS C:\Users\Windows> Write-host "El nombre es: $archivo.name"
El nombre es: C:\Users\Windows\prueba.txt.name
PS C:\Users\Windows> Write-host "El nombre es: $($archivo.name)"
El nombre es: prueba.txt
```

6.- Lectura y escritura de una variable en un script.

Práctica:LecturaEscritura.ps1

Entrada:

Pregunta el nombre de usuario y la contraseña.

La contraseña se debe cifrar.

Salida:

Muestra el nombre de usuario y la contraseña introducidos.

6.- Lectura y escritura de una variable en un script.

```
LecturaEscritura.ps1 X
1  #Descripción
2  #Nombre:
3  #Autor: Manuel Domínguez
4  #Fecha: 16/11/2019
5  #Versión: 1.0
6  #Definición de parámetros
7  #Definición de funciones
8  #Bloque principal
9  Clear-Host
10 #Leemos dos datos por pantalla
11 $usuario= Read-Host "Introduce tu nombre de usuario"
12 $contra= Read-Host "Introduce la contraseña" -AsSecureString
13 #Mostramos los datos introducidos
14 Write-Host "Usuario: $usuario"
15 Write-Host "Contraseña: $contra"
16
```

```
<
Introduce tu nombre de usuario: manolo
Usuario: manolo
Contraseña: System.Security.SecureString

PS C:\Users\Windows> |
```

Resumen

Variable: espacio de la memoria principal a la que le asignamos un nombre para poder acceder.

Empiezan por el símbolo \$

No es obligatorio declararla.

Constante: Una variable cuyo valor no cambia.

Nombre en mayúsculas.

Utilizaremos el cmdlet: New-Variable, con la Option Constant

Contenido de una variable/constante: \$Nombre

Averiguar el tipo de una variable: GetType()

Cmdlets para trabajar con variables: Clear-Variable, **Get-Variable**, New-Variable, Remove-Variable, Set-Variable.

Read-Host y Write-Host: Lectura y escritura de variables en un script.

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.