

# 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.

## 1.- Operadores

Una de las tareas más comunes es la de realizar operaciones con las variables.

Existen muchos tipos de operadores, pero podemos destacar:

- Aritméticos.

- De asignación.

- De comparación.

- Lógicos

- De tipos

## 2.- Operadores aritméticos.

Operador	Descripción	Ejemplo
+	Suma	$\$suma = \$x + \$y$
-	Resta	$\$resta = \$x - \$y$
*	Multiplicación	$\$multi = \$x * \$y$
/	División	$\$divi = \$x / \$y$
%	Módulo (Resto) División entera	$\$modulo = \$x \% \$y$

## 2.- Operadores aritméticos.

### Práctica:

Realizar un script: **OperAritmeticos.ps1**

Entrada: Dos variables decimales x e y

Salida: Muestra en pantalla: +, -, \*, / y %  
Recuerda que % es una división entera.

## 2.- Operadores aritméticos.

OperAritmeticos.ps1 X

```
7  #Definición de funciones
8  #Bloque principal
9  Clear
10 Write-Host "Operaciones aritméticas `n`" "#Salto
11 #De momento no controlamos los errores en la entrada de datos.
12 #Declaramos las variables como double (Decimales)
13 [double]$x=Read-Host "Introduzca el valor de x"
14 [double]$y=Read-Host "Introduzca el valor de y"
15 #Realizamos las operaciones
16 [double]$suma=$x + $y
17 [double]$resta=$x - $y
18 [double]$multi=$x * $y
19 [double]$divi=$x / $y
20 #El módulo es el resto de una división entera.
21 [int]$modulo=[int]$x % [int]$y
22 #Mostramos los resultados
23 Write-Host "La suma es: $suma"
24 Write-Host "La resta es: $resta"
25 Write-Host "La multiplicación es: $multi"
26 Write-Host "La división es: $divi"
27 Write-Host "El módulo es: $modulo"
```

### Operaciones aritméticas

Introduzca el valor de x: 20.2  
Introduzca el valor de y: 3  
La suma es: 23.2  
La resta es: 17.2  
La multiplicación es: 60.6  
La división es: 6.73333333333333  
El módulo es: 2

## 2.- Operadores aritméticos: Clase math de .NET

PowerShell está basado en .NET Framework.

Podemos utilizar la clase llamada **math**, que nos proporciona un abanico de operaciones matemáticas.

Para ver qué métodos incorpora, podemos escribir:

**[math].GetMethods() |Ft Name**

Entre otros están: floor, ceiling, Round, sqrt, Abs, Log, etc.

Para utilizarla: **[math]::Método (\$variable o valores)**

**Ejemplos:** [math]::sqrt(9)      \$x=3.5 → [math]::floor (\$x)



## 2.- Operadores aritméticos: Clase math de .NET

### Práctica:

Realizar un script: **CalculoEdad.ps1**

**Entrada:** La fecha de nacimiento.

**Salida:** Nos muestra la edad actual.

**Pista:** [datetime], Get-date, [math]::floor

## 2.- Operadores aritméticos: Clase math de .NET

CalculoEdad.ps1 X

```
1  #Descripción
2  #Nombre:CalculoEdad.ps1
3  #Autor: Manuel Domínguez
4  #Fecha: 17/11/2019
5  #Versión: 1.0
6  #Definición de parámetros
7  #Definición de funciones
8  #Bloque principal
9  Clear
10 Write-Host "Calculo de la edad"
11 [datetime]$nacimiento=Read-Host "Introduce tu fecha de nacimiento (MM/DD/AAAA)"
12 [datetime]$actual=Get-date
13 #Calculamos la diferencia entre las fechas. Nos da la diferencia en: Days, hours,Minute, etc
14 $Diferencia= $actual - $nacimiento
15 #Nos quedamos con los días y calculamos los años dividiendo por 365
16 $anos=($diferencia.Days/365)
17 #Nos quedamos con el número entero mayor, pero menor a x.
18 [math]::floor($anos)
```

## 2.- Operadores aritméticos.

### **Reto:**

Realizar un script: **OperMatematicas.ps1**

Entrada: Dos variables enteras x e y

Salida: La potencia de x elevado a Y

**Pista:** Investiga qué **método de la clase .Net** podemos utilizar.

**Solución reto:** Intenta resolverlo sin mirar la solución que se muestra a continuación.

## Solución al reto.

### Reto: OperMatematicas.ps1

```
OperMatematicas.ps1* X
4  #Fecha: 16/11/2019
5  #Versión: 1.0
6  #Definición de parámetros
7  #Definición de funciones
8  #Bloque principal
9  Clear
10 Write-Host "Operaciones Matemáticas con la clase:
11 [Math] de .NET"
12 #De momento no controlamos los errores en la entrada de datos.
13 #Declaramos las variables como entera
14 [int]$x=Read-Host "Introduzca el valor de x"
15 [int]$y=Read-Host "Introduzca el valor de y"
16 #Realizamos la operación llamando a clase [System.Math] de .NET
17 [int]$potencia=[Math]::Pow($x,$y)
18 #Mostramos los resultados
19 Write-Host "La potencia es: $potencia"
```

### 3.- Operadores de asignación.

Estos operadores, nos van a permitir al mismo tiempo, realizar una operación y asignar el resultado a una variable.

Operador	Descripción	Ejemplo
=	Asignar un valor	\$x=10
+=	Suma a la variable el valor indicado.	\$x+=5 (\$x=\$x+5)
-=	Resta a la variable el valor indicado.	\$x-=5 (\$x=\$x-5)
*=	Multiplica la variable por el valor indicado.	\$x*=5 (\$x=\$x*5)
/=	Divide la variable por el valor indicado.	\$x/=5 (\$x=\$x/5)

### 3.- Operadores de asignación.

Los siguientes operadores se utilizan mucho en las condiciones de los bucles.

Operador	Descripción	Ejemplo
++	Suma a la variable el valor 1.	$x++$ ( $x = x + 1$ )
--	Resta a la variable el valor 1.	$x--$ ( $x = x - 1$ )

# 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.