

Estructura condicional If, Else, ElseIf

Una estructura condicional permite, mediante una evaluación de la condición, orientar la ejecución de las instrucciones. La sintaxis de una estructura condicional es la siguiente:

```
If (condición)
{
    #bloque de instrucciones
}
```

Para comprender mejor la utilización de una estructura condicional, he aquí algunos ejemplos de sus aplicaciones:

Supongamos que queremos determinar si un valor entrado por el usuario es la letra «A». Para ello, vamos a utilizar una estructura condicional con una condición sobre el valor de la variable verificada. Utilizando un operador de comparación, la estructura será la siguiente:

```
$var = Read-Host

If($var -eq 'A')
{
    Write-Host "El carácter introducido por el usuario es una 'A'"
}
```

Si la variable entrada por el usuario es una «A», entonces el commandlet `write-host` será tratado, sino, la ejecución proseguirá su curso.

En la instrucción `If`, puede asociarse la cláusula `Else`. Esta cláusula, permite, en caso de devolver un valor `False`, dirigir el tratamiento hacia un segundo bloque de instrucciones. Según el ejemplo siguiente:

```
If (($var1 -eq 15) -and ($var2 -eq 18))
{
    # Bloque de instrucciones 1
}
Else
{
    # Bloque de instrucciones 2
}
```

En primer lugar, PowerShell evalúa la primera condición, para saber si la variable `$var1` es igual a 15. Si la verificación es buena entonces la primera condición toma el valor `true`.

Después, evalúa la segunda condición (`$var2 -eq 18`). Si la verificación es buena entonces la segunda condición `$var2` toma el valor `true`.

Si los dos valores son verdaderos, el operador lógico `-and` de la condición devuelve el valor `true` (verdadero Y verdadero = verdadero), y por tanto se ejecuta el bloque de instrucción 1, por el contrario, si la condición es falsa, se ejecuta el bloque de instrucción 2.

Siguiendo en el mismo registro veamos otro ejemplo:

```
[int]$var1 = Read-Host 'Introduzca un número'
[int]$var2 = Read-Host 'Introduzca un número'

If($var1 -ge $var2)
{
    Write-Host "$var1 es mayor o igual que $var2"
}
Else
```

```
{  
    Write-host "$var1 es menor que $var2"  
}
```

En este segundo ejemplo, el usuario inicializa dos variables, y luego PowerShell comprobará si el primer valor es mayor o igual al segundo. Si la condición es verdadera, entonces en la consola se muestra un mensaje indicando que el primer valor es mayor o igual al segundo. Si condición es falsa, se muestra el mensaje situado en el bloque de instrucción de la cláusula ELSE.

Por último, para finalizar con las estructuras condicionales, aquí tiene cómo mejorarlas con la instrucción `ElseIf`. La instrucción `ElseIf` nos permitirá, si la condición anterior es falsa, verificar otra condición. De este modo, utilizando una estructura condicional con `ElseIf`, no nos limitaremos a una orientación binaria, sino que aumentaremos las posibles orientaciones de ejecución.

Ejemplo:

```
[int]$val = read-host 'Introduzca un valor: 1,2 o 3'  
  
If($val -eq 1)  
    { Write-Host 'el valor introducido es igual a 1 '  
ElseIf($val -eq 2)  
    { Write-Host 'el valor introducido es igual a 2 '  
ElseIf($val -eq 3)  
    { Write-Host 'el valor introducido es igual a 3 '  
Else  
    {Write-Host "el valor introducido no es igual ni a 1, ni a 2, ni a 3"}
```

De esta forma, se podrán crear tantos `ElseIf` como deseemos. Sin embargo, la utilización intensa de `ElseIf` es una solución viable pero un poco pesada. El hecho de que existan tantas condiciones como bloques de instrucción, hace el código poco flexible, siendo preferible decantarse por la instrucción `Switch`.