Switch

La instrucción `Switch` permite reemplazar toda una serie de `If` y de `ElseIf`. A diferencia de la instrucción `If` que, para una expresión dada, orientará el conjunto de la ejecución hacia uno de los dos bloques de instrucciones, la instrucción `Switch` nos permite orientar la ejecución hacia varios bloques de instrucciones. Y todo ello, con una única expresión, lo que le confiere una utilización mucho más flexible. La sintaxis de `Switch` es la siguiente:

```
Switch (<Expresión>)
{
        <Valor_1> {<instrucciones>}
        <Valor_2> {<instrucciones>}
        <Valor_3> {<instrucciones>}
        Default   {<instrucciones>}
}
```

El valor « default » es opcional, su bloque de instrucción se ejecuta únicamente en el caso en que la expresión no corresponda a ninguno de las valores de `Switch`.

Tomemos como ejemplo de aplicación, el caso básico donde el usuario debe introducir un número entre 1 y 5, y PowerShell determina qué número se ha introducido. El código es el siguiente:

```
$Numero = Read-Host 'Introduzca un número comprendido entre 1 y 5 '

Switch($Numero)
{
   1 {Write-Host 'Ha introducido el número 1 '}
   2 {Write-Host 'Ha introducido el número 2 '}
   3 {Write-Host 'Ha introducido el número 3 '}
   4 {Write-Host 'Ha introducido el número 4 '}
   5 {Write-Host 'Ha introducido el número 5 '}
   Default {Write-Host "El número introducido no está comprendido entre 1 y 5"}
}
```

La instrucción `Switch` acepta igualmente las expresiones regulares, para ello basta con especificar el parámetro `-regex`:

```
$cadena = Read-Host 'Introduzca una cadena'

Switch -regex ($cadena)
{
   '^[aeiouy]'  {Write-Host 'La cadena introducida empieza por una vocal'}
   '^[^aeiouy]' {Write-Host 'La cadena introducida no empieza por una vocal'}
}
```

Si existen varias correspondencias, cada una de ellas provoca la ejecución del bloque de instrucción correspondiente. Para evitar esto, utilice la palabra clave `break` que permite ejercer una salida de ejecución.