

Creación de objetos personalizados

Como acabamos de ver anteriormente, devolver un valor al final de función o de script es muy sencillo. En el caso de los scripts consecuentes con muchos resultados de retorno, la visualización de los resultados sin un formateo particular no es siempre fácil de interpretar ya que no garantiza una uniformidad de los resultados. Más exactamente, ¿qué significa esto? Tomamos el ejemplo del script siguiente que, para un determinado archivo, devuelve su nombre, la fecha de creación y la fecha del último acceso:

```
Function Info-File
{
    param([string]$nombre_archivo)
    $archivo = Get-Item $mi_archivo
    Write-Output "Nombre: $($archivo.Name)"
    Write-Output "Fecha de creación: $($archivo.CreationTime)"
    Write-Output "Último acceso: $($archivo.LastAccessTime)"
}
```

El resultado de esta función es una lista de los diferentes archivos con las propiedades solicitadas.

```
PS > Info-File .\Abonados.txt

Nombre: Abonados.txt
Fecha de creación: 04/09/2009 18:17
Último acceso: 04/09/2009 18:17:00
```

Si bien el resultado es el esperado, su interpretación y su reutilización no es del todo satisfactoria. Es preferible utilizar un formato diferente en forma de objeto personalizado. Un objeto personalizado es un objeto al que se le va a poder insertar nuestras propias propiedades para adoptar un formato claro y reutilizable. Veamos un ejemplo con el siguiente script que realiza el mismo tratamiento que el anterior.

```
Function Info-File
{
    param([string]$nombre_archivo)
    $archivo = Get-Item $nombre_archivo
    # Construcción del objeto que contendrá todos los resultados
    $resultado = New-Object PSObject
    # Añadir los miembros a nuestro objeto:
    $resultado | Add-Member NoteProperty Nombre $archivo.Name
    $resultado | Add-Member NoteProperty Fecha_Creacion $archivo.CreationTime
    $resultado | Add-Member NoteProperty Ultimo_Acceso $archivo.LastAccessTime

    # Devuelve los resultados
    $resultado
}
```

Esta vez hemos creado nuestro propio objeto PowerShell (véase el Capítulo Control del Shell - Objetos PSBase y PSObject) al que hemos añadido propiedades gracias al comando **add-member**. A cada una de estas propiedades se le asignará un valor. Así, cuando el objeto es devuelto, adoptará la forma de una tabla en la que cada columna es una propiedad.

```
PS > Info-File .\Abonados.txt

Nombre          Fecha_Creacion      Ultimo_Acceso
---          -
Abonados.txt    04/09/2009 18:17:00 04/09/2009 18:17:00
```

Es por tanto mucho más sencillo leer la información y reutilizar el objeto posteriormente.

```
PS > $resultado = Info-File .\Abonados.txt
PS > $resultado.Nombre
Abonados.txt

PS > $resultado.Ultimo_Acceso
04/09/2009 18:17:00
```

Otro ejemplo; tomemos esta vez el ejemplo de un script que muestra para un directorio determinado, el número y los nombres de los scripts cuyo tamaño está comprendido entre 0 y 50 bytes, 50 y 500 bytes, y más de 500 bytes.

```
#Information-tamaño.ps1
param([string]$ruta)

#Creación de las variables
$lista = Get-ChildItem $ruta
$Tamaño_0_50 = 0
$lista_0_50 = ''
$Tamaño_50_500 = 0
$lista_50_500 = ''
$Tamaño_500 = 0
$lista_500 = ''

#Bucle sobre el conjunto de los elementos contenidos en la lista
foreach($elemento in $lista){
    $type=$elemento.gettype()
    $size=0
    if($type.Name -eq "FileInfo"){
        $size=$elemento.length
        if($size -lt 50){
            $Tamaño_0_50++
            $lista_0_50 += "$elemento;"
        }
        elseif(($size -ge 50) -and ($size -lt 500)){
            $Tamaño_50_500++
            $lista_50_500 += "$elemento;"
        }
        elseif($size -ge 500){
            $Tamaño_500 ++
            $lista_500 += "$elemento;"
        }
    }
}

# Construcción del objeto que contendrá todos los resultados asociados
# a los archivos de tamaño 0 a 50 bytes
$Result_0_50 = New-Object PSObject
# Adición de los miembros a nuestro objeto:
$Result_0_50 | Add-Member NoteProperty 'Tamaño (bytes)' '0 - 50'
$Result_0_50 | Add-Member NoteProperty Numero $Tamaño_0_50
$Result_0_50 | Add-Member NoteProperty Nombres $lista_0_50

# Construcción del objeto que contendrá todos los resultados asociados
# a los archivos de tamaño 50 a 500 bytes
```

```

$Result_50_500 = New-Object PSObject
# Adición de los miembros a nuestro objeto:
$Result_50_500 | Add-Member NoteProperty 'Tamaño (bytes)' '50 - 500'
$Result_50_500 | Add-Member NoteProperty Numero $Tamaño_50_500
$Result_50_500 | Add-Member NoteProperty Nombres $lista_50_500

# Construcción del objeto que contendrá todos los resultados asociados
# a los archivos de tamaño 500 bytes y más
$Result_500 = New-Object PSObject
# Adición de los miembros a nuestro objeto:
$Result_500 | Add-Member NoteProperty 'Tamaño (bytes)' '500+'
$Result_500 | Add-Member NoteProperty Numero $Tamaño_500
$Result_500 | Add-Member NoteProperty Nombres $lista_500

#visualización de los objetos personalizados
$Result_0_50
$Result_50_500
$Result_500

```

La construcción de los objetos personalizados y su visualización en forma de tabla permite obtener un resultado claro y preciso.

```
PS .\Informacion_tamaño.ps1 D:\Scripts
```

Tamaño (bytes)	Número	Nombres
-----	-----	----
0 - 50	2	Prueba.txt;Script.txt;
50 - 500	1	cadena_c2.txt;
500+	6	cesar.ps1;cadena_cl.txt;cadena_...