

# Understanding Objects in Windows PowerShell

---



**Jeff Hicks**

Author/Teacher

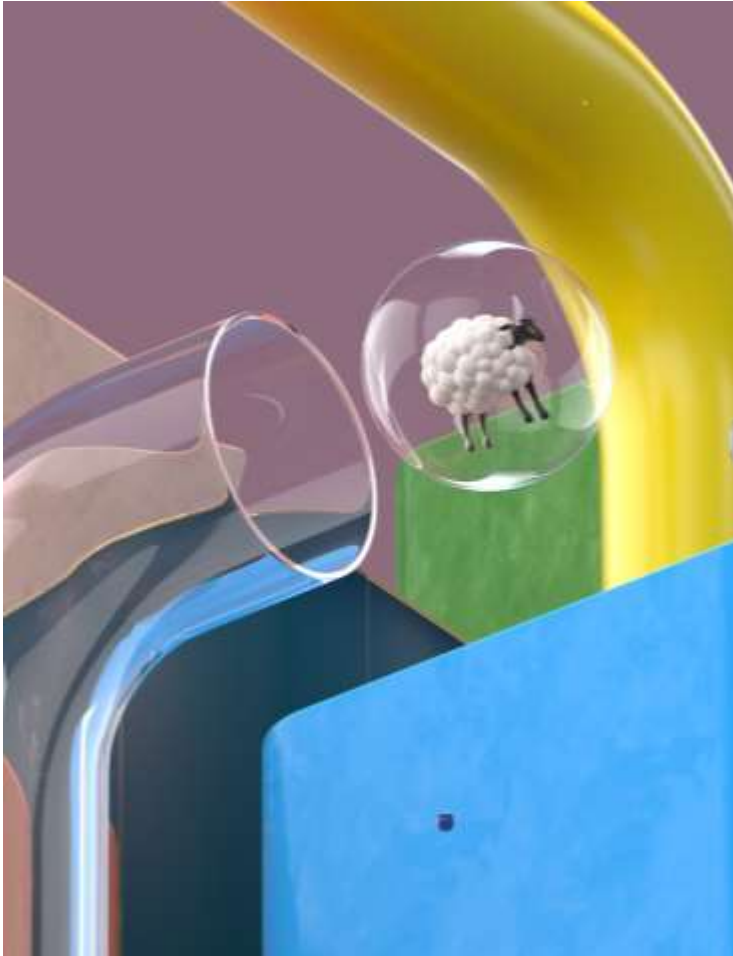
@jeffhicks | <https://jdhitsolutions.com/blog>



Windows PowerShell is an interactive management tool that works with objects, not text, in a pipeline.



# Objects in the Pipeline



**Traditional management and scripting tools are text based**

**Manipulate and parse text for results**

**Windows PowerShell designed to manage Windows and Microsoft platforms with .NET**

**It is all about the objects**





**Windows PowerShell commands create and (optionally) consume objects**

**They are passed to the next command in the pipeline**

**Objects might change in the pipeline**

**At the end of the pipeline, PowerShell displays whatever is left**





An object is a software representation of the *thing* you want to manage

Describe the object with *Properties*

- Color
- Count
- SKU

Perform an action or be acted upon from *Methods*

- Peel
- Chop
- Roast

You typically use cmdlets to interact with properties and methods



PS C:\>

# Windows PowerShell Commands and Objects

***Everything* in Windows PowerShell is some type of object**



```
PS C:\> Get-Service -Name bits
```

# Windows PowerShell Commands and Objects

***Everything* in Windows PowerShell is some type of an object**

**Parameter values**



```
PS C:\> Get-Service -Name bits
```

Status	Name	DisplayName
-----	----	-----
Running	bits	Background Intelligent Transfer Servi...

# Windows PowerShell Commands and Objects

***Everything*** in Windows PowerShell is some type of an object

**Parameter values**

**Command output**





# Common Types

**String**

**Int**

**DateTime**

**Boolean**





## **View objects with Get-Member**

- Has an alias of gm

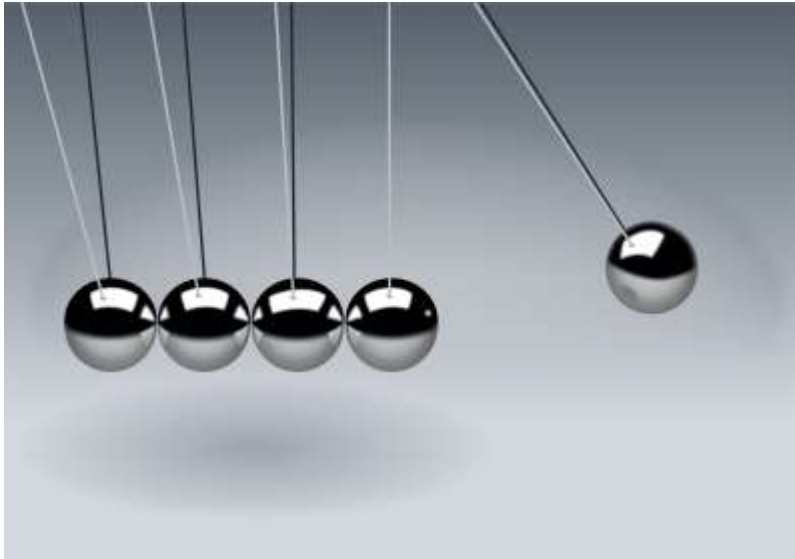
**Don't assume what you see on the screen is all there is**

**You can use any property you discover with Get-Member**

**Not getting the result you expect?**

**Pipe your command to Get-Member**





## PowerShell commands write objects to the pipeline

- Although this isn't a requirement
- -PassThru

## Some commands can consume objects

- Use the entire object
- Use a property value

Think “Manage at Scale”



```
PS C:\> Get-Service bits,spooler
```

## Pipeline Expression

**Get bits and spooler services**



```
PS C:\> Get-Service bits,spooler |
```

## Pipeline Expression

**Get bits and spooler services**

**Send them to another command in the pipeline**



```
PS C:\> Get-Service bits,spooler | Set-Service -startuptype manual
```

## Pipeline Expression

**Get bits and spooler services**

**Send them to another command in the pipeline**

**Do an action on the service objects**



```
PS C:\> Get-Service bits,spooler | Set-Service -startuptype manual  
PS C:\>
```

## Pipeline Expression

**If no other commands, the pipeline finishes**

**PowerShell will write remaining objects to the console – if any.**



# Demo



## Objects in the Pipeline

