Windows **PowerShell**

PROGRAMMING PARADIGMS

(2ND SEMESTER, 2014/2015)

João Pedro Dias (ei11137@fe.up.pt)
Hugo Freixo (ei11086@fe.up.pt)

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

WINDOWS POWERSHELL

- Developed by Microsoft for Windows Operative System's.

- Is a task-based command-line shell and scripting language.
  - Appeared as an update for the pre-existent command-line interface tool *cmd.exe*

- It is designed especially for system administration that is used by information technology professionals on a regular basis.

WINDOWS POWERSHELL

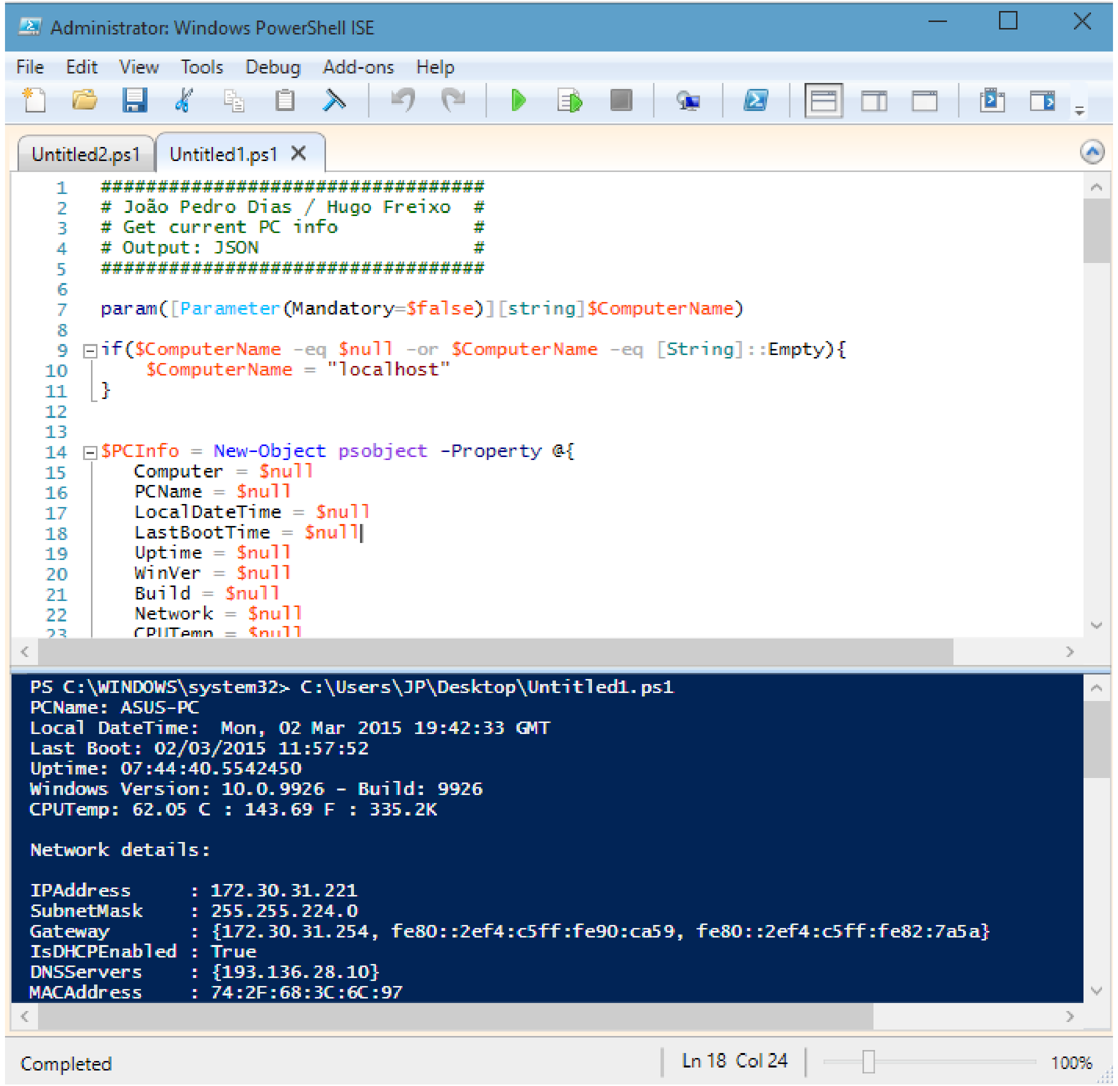
- Built in C++ language.
- Is built on the top of .NET framework.
- Can execute:
  - *cmdlets* – "*command let*" (.NET programs that interact with PowerShell)
  - PowerShell *scripts* (.ps1 files)
  - PowerShell *functions*
  - Standalone executable programs

Examples of cmdlet:

- Get-Location
- Set-Location
- Copy-Item
- Remove-Item
- Move-Item
- Rename-Item
- New-Item

WINDOWS POWERSHELL ISE

- *Windows PowerShell Integrated Scripting Environment (ISE)* is a tool for development and run PowerShell scripts (.ps1).



```
###################################
# João Pedro Dias / Hugo Freixo  #
# Get current PC info             #
# Output: JSON                    #
###################################

param([Parameter(Mandatory=$false)][string]$ComputerName)

if($ComputerName -eq $null -or $ComputerName -eq [String]::Empty){
    $ComputerName = "localhost"
}


$PCInfo = New-Object psobject -Property @{
    Computer = $null
    PCName = $null
    LocalDateTime = $null
    LastBootTime = $null
    Uptime = $null
    WinVer = $null
    Build = $null
    Network = $null
    CPUTemp = $null
```

```
PS C:\WINDOWS\system32> C:\Users\JP\Desktop\Untitled1.ps1
PCName: ASUS-PC
Local DateTime:  Mon, 02 Mar 2015 19:42:33 GMT
Last Boot: 02/03/2015 11:57:52
Uptime: 07:44:40.5542450
Windows Version: 10.0.9926 - Build: 9926
CPUTemp: 62.05 C : 143.69 F : 335.2K

Network details:

IPAddress     : 172.30.31.221
SubnetMask    : 255.255.224.0
Gateway       : {172.30.31.254, fe80::2ef4:c5ff:fe90:ca59, fe80::2ef4:c5ff:fe82:7a5a}
IsDHCPEnabled : True
DNSServers    : {193.136.28.10}
MACAddress    : 74:2F:68:3C:6C:97
```

HISTORY

PowerShell 1.0

Initial release: 2006 (9 years ago)

- Released for Windows XP SP2, Windows Server 2003 and Windows Vista.

PowerShell 2.0

Released in 2009

- PowerShell Remoting
- Background Jobs
- Modules
- Script Debugging
- Windows PowerShell ISE
- Network File Transfer

PowerShell 3.0

Released in 2012

- Schedule jobs
- Session connectivity
- Task delegation support

PowerShell 4.0

Released in 2013

- Where and ForEach
- Network diagnostics
- Desired State Configuration

PowerShell 5.0

Released in 2014

- Develop using classes (Class, Enum)
- PowerShellGet (PowerShell Resource Gallery)

POWERSHELL VS. WINDOWS BATCH SCRIPTING

- Cmdlets
  - Managing the registry or WMI (Windows Management Instrumentation).
- Use of pipes.
- Powerful scripting environment.
- Almost full access to .NET framework functionalities.

- Legacy environment carried forward in Windows
  - An environment that copies all of the various DOS commands you would find on a DOS system.
- Limited scripting functions
  - May contain any command the interpreter accepts interactively at the command prompt plus Goto, For and If.

POWERSHELL VS. LINUX SHELL SCRIPTING

- PowerShell command window has support for a ton of the *nix shell commands.
  - ls, pwd, etc.
- Everything is an object. You can pass the result of a command to another as an object.

- Everything is a file.
  - The input and output of a Unix command can be accessed like a file. This makes chaining command really simple and reusing the output of one command by another is the power of the Unix shell.

POWERSHELL ANATOMY OBJECT-BASED

- One of the potential (dis)advantages of using Windows PowerShell is that it is object-based.
- With most shells, you rely on text-based commands to get the job done when writing scripts. If you switch to Windows PowerShell from some other type of shell, you'll have to get used to a different way of thinking about things.
- This can be a problem that takes some time to get past for some users.

#!/bin/bash

POWERSHELL ANATOMY SECURITY RISKS

- Many IT professionals use it as a way to connect remotely to other computers and servers.
- When engaging in this process, PowerShell can leave some holes open for security breaches.
- This creates a potential for viruses, malware or other harmful programs to be installed in the server.

POWERSHELL ANATOMY WEB SERVER

- Another issue with Windows PowerShell is that it requires you to run a Web server on your server when utilizing remote functionality.
- This takes up additional space on a server. In many cases, companies will not want to take up more room and designate more resources to this on their own servers.

POWERSHELL ANATOMY ADVANTAGES

- Since it is developed by Microsoft, it is being integrated more and more into Microsoft products and services.
- Windows PowerShell is also versatile and easy to administrate once you learn the basics of the scripting language.
- It also gives you the ability to run specific commands that are designed to run only on local networks if you are using the remote connection function.

Windows 10

Windows 8

Windows 7

Windows Vista™

Microsoft® Windows xp

POWERSHELL SYNTAX

Variables

```
$var = 'hello‘
$number = 1
$numbers = 1,2,3,4,5,6,7,8,9
$name = 'Don'
$prompt = "My name is $name”

PS C:\WINDOWS\System32> $prompt
My name is Don
```

Object Members and Variables

```
$var = 'Hello'
$var | Get-Member

$svc = Get-Service
$svc[0].name
$name = $svc[1].name
$name.length
$name.ToUpper()
```

POWERSHELL SYNTAX

If Construct

```
If ($this -eq $that) {
  # commands
} elseif ($those -ne $them) {
  # commands
} elseif ($we -gt $they) {
  # commands
} else {
  # commands
}
```

Do While Construct

```
Do {
  # commands
} While ($this -eq $that)

While (Test-Path $path) {
  # commands
}
```

POWERSHELL SYNTAX

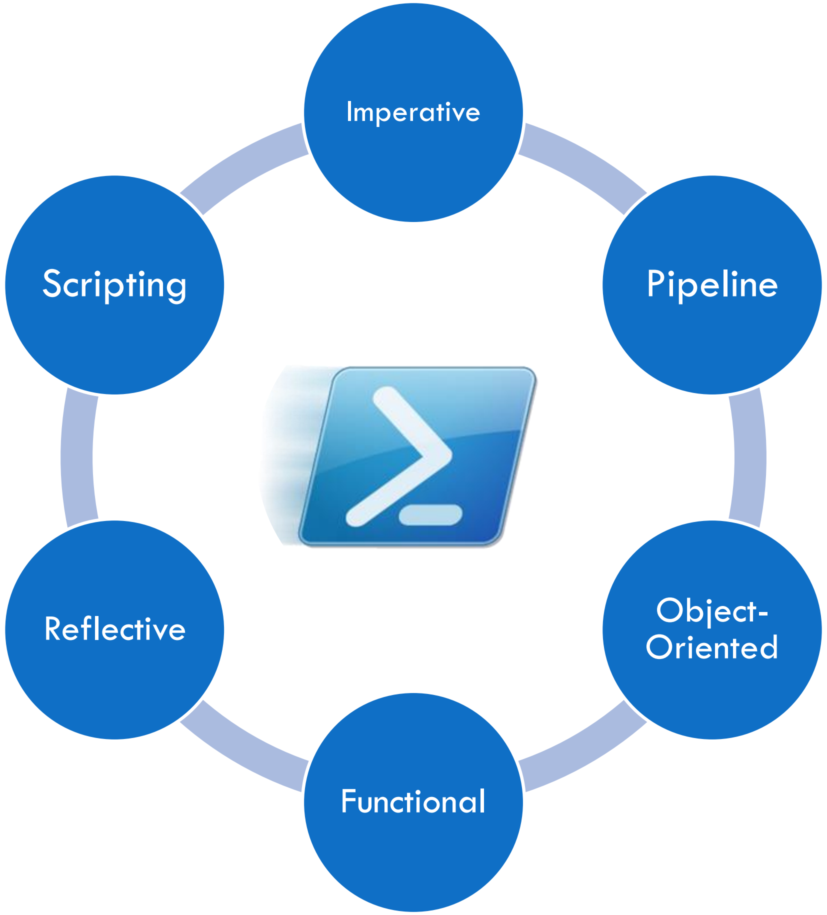
ForEach Construct

```
$services = Get-Service

ForEach ($service in $services)
{
   $service.Stop()

}
```

Functions

```
function func {

   Get-Service

   Get-Process

}

func
```

PROGRAMMING PARADIGMS



SCRIPTING PARADIGM

```
$t = Get-WmiObject MSAcpi_ThermalZoneTemperature -
Namespace "root/wmi“
$result = @()
foreach ($temp in $t.CurrentTemperature)
{
 $TempK = $temp / 10
 $TempC = $TempK - 273.15
 $TempF = (9/5) * $TempC + 32
 $result += $TempC.ToString() + “C`n" +
   $TempF.ToString() + “F`n" + $TempK + "K"
}
Write-Host $result
```

- *“Scripting Language: (skript´ing lang´gwij)* ***(n.)*** *A high-level programming language that is interpreted by another program at runtime rather than compiled by the computer's processor as other programming languages (such as C and C++) are.”*

in Webopedia

IMPERATIVE PARADIGM

```
Import-Module ServerManager

#Check and install ASP.NET 4.5 feature

If (-not(Get-WindowsFeature "Web-Asp-Net45").Installed)
{
   try {
         Add-WindowsFeature Web-Asp-Net45
   }
   catch {
         Write-Error $_
   }
}
```

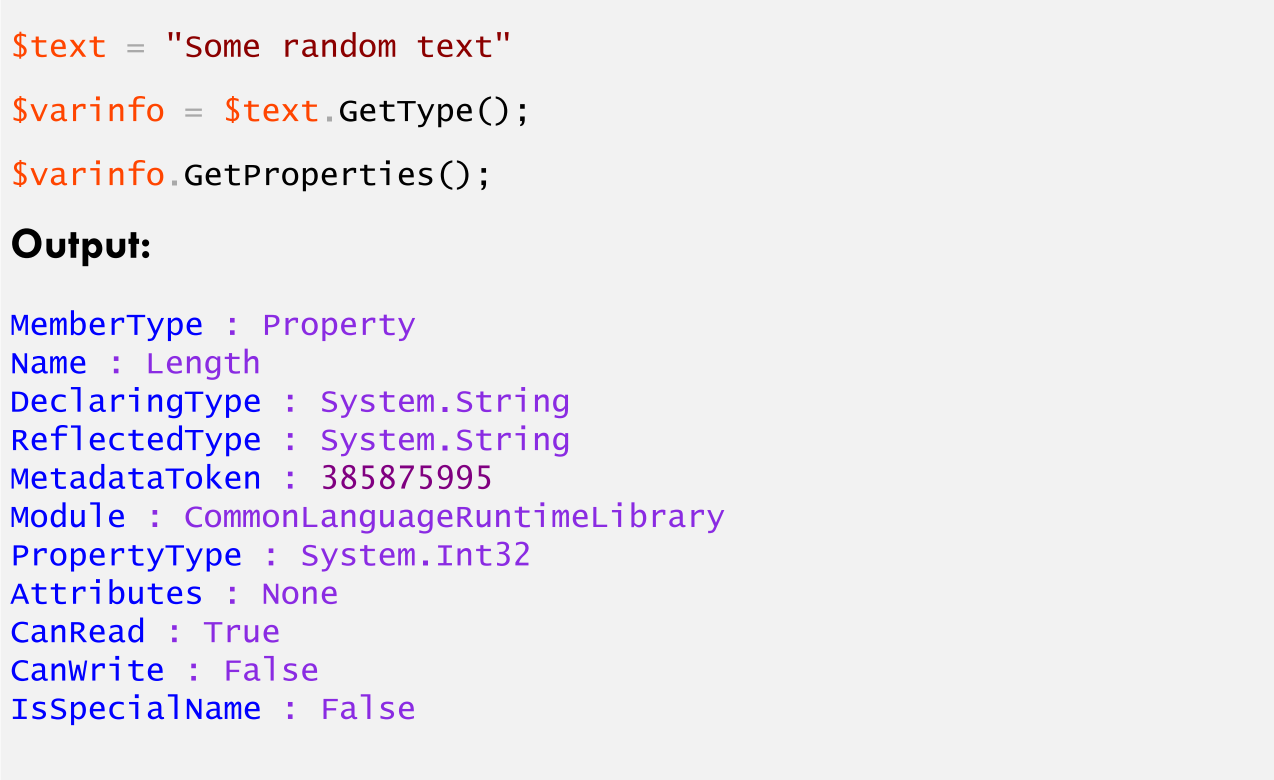
- "Computer, add x and y“
- "Computer, open a dialog box onto the screen."
- “Computer, check if the *ASP.NET* is installed, if not, install it."

OBJECT-ORIENTED PARADIGM

```
$DogClass = new-object psobject -Property @{
  color = $null
  name = $null
  _size = $null
}

$dogclass |Add-Member -PassThru -MemberType ScriptMethod -Name
Size -Value {
  param(
       [Parameter(Mandatory=$false, Position=0)]
       $Size
  )
  if ($size) {
     $this._size = $size
  } else {
     $this._size
  }

}
```

- Class
  - Constructors
  - Methods
- Inheritance
  - Overriding

REFLECTIVE PARADIGM



```
$text = "Some random text"

$varinfo = $text.GetType();

$varinfo.GetProperties();
```

Output:

```
MemberType : Property
Name : Length
DeclaringType : System.String
ReflectedType : System.String
MetadataToken : 385875995
Module : CommonLanguageRuntimeLibrary
PropertyType : System.Int32
Attributes : None
CanRead : True
CanWrite : False
IsSpecialName : False
```

- "(…)ability to observe or change its own code as well as all aspects of its programming language (syntax, semantics, or implementation) at runtime."

  in RosettaCode

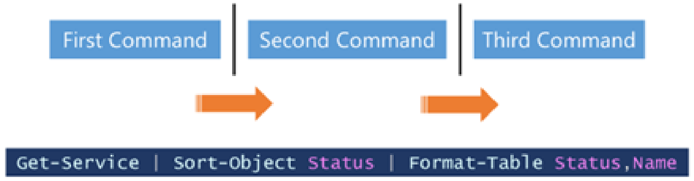
PIPELINE PARADIGM

```
 # Simple PowerShell Pipeline Example

Get-Process | Where-Object {$_.handlecount -gt 100 }


#The problem: We need to control the properties

#The solution: A second pipe, then control the display
with Format-Table


Get-Process `

| Where-Object {$_.company -Notlike '*Microsoft*'}`

| Format-Table ProcessName, Company -auto
```

UNIX®

- Origins remote to the Unix based Operative Systems.
- *"The pipe character is used between commands to create the pipeline. We work from left to right down the pipeline. The output of one command effectively becomes the input of the next command."*

in http://blogs.technet.com



```
Get-Service | Sort-Object Status | Format-Table Status,Name
```

FUNCTIONAL PARADIGM

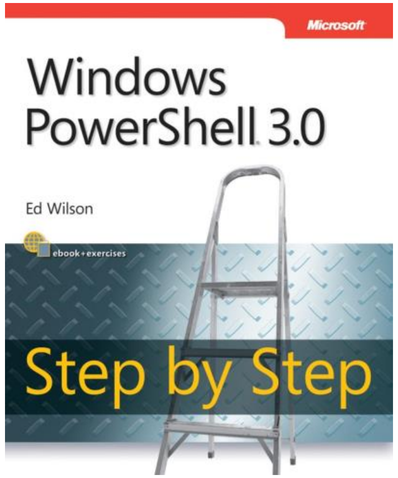
```
 # Immutable
function New-ImmutableObject($object) {
 $immutable = New-Object PSObject
    $object.Keys | % {
        $value = $object[$_]
        $closure = { $value }.GetNewClosure()
        $immutable | Add-Member -name $_ -memberType
ScriptProperty -value $closure
    }
return $immutable
}

#Higher Order Functions
function Convert-ByFilter($values, $predicate) {
  return $values | where { & $predicate $_ }
}
```

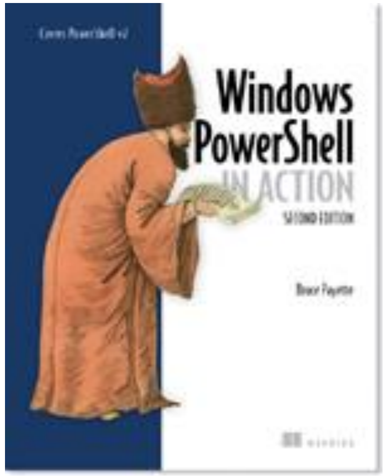
- Immutable Object
- Higher Order Functions
- Currying
- Lazy Evaluation (*System.Lazy*)
- Pattern Matching
- Closures (*GetNewClosure*)

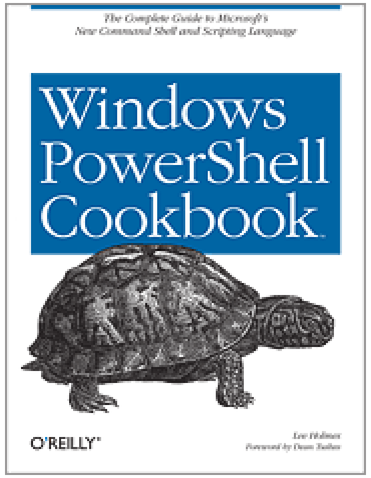
U.PORTO

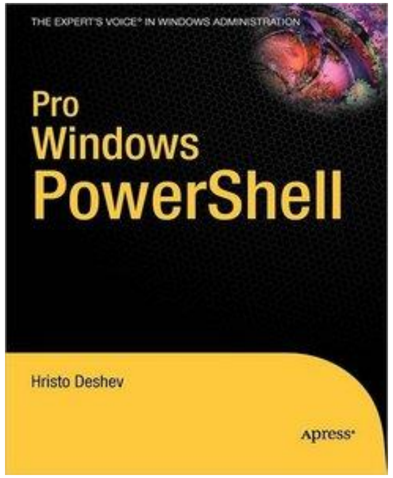
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

REFERENCES






The Complete Guide to Microsoft's New Command Shell and Scripting Language
Windows PowerShell Cookbook
O'REILLY
Lee Holmes



- Windows PowerShell 3.0 Step by Step (Step by Step Developer) by Ed Wilson
- Windows PowerShell Cookbook: The Complete Guide to Scripting Microsoft's Command Shell by Lee Holmes
- Windows PowerShell in Action, Second Edition by Bruce Payette
- Pro Windows PowerShell by Hristo Deshev

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

ONLINE RESOURCES

- IRC - #powershell on freenode
- Reddit - /r/powershell
- News
  - http://www.powershellmagazine.com/
- Blogs & Websites
  - http://blogs.msdn.com/b/powershell/
  - http://powershell.com/
- Scripts Repositories
  - http://gallery.technet.microsoft.com/ScriptCenter/
  - http://technet.microsoft.com/en-us/scriptcenter/default.aspx
  - http://poshcode.org/

Windows **PowerShell**

THANK YOU! | Questions?

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO