

# CSI3670

## A Bit More on PowerShell

---

Erik Fredericks ([fredericks@oakland.edu](mailto:fredericks@oakland.edu))

# More PowerShell

## PLAN OF ACTION

- Couple of 'minor' points about PowerShell
- Small amount of labwork
- More term project time!

Tuesday:

- In-class work (bring your laptops)
  - Won't be graded, but worthwhile

# Functions vs. *Advanced* Functions

```
Function Do-Something {  
    Param($item)  
    Write-Host "You passed in  
$item"  
}
```

```
Do-Something -item "aaa"
```

```
Function Do-Something {  
    [CmdletBinding()]  
    Param($item)  
    Write-Host "You passed in  
$item"  
}
```

# Advanced Functions

Not necessarily a cmdlet, but takes advantage of their features

Such as:

- Verbosity
- Pipeline (object transfer)
- WhatIf
- ...much more

# Verbosity

Verbosity: the act of making things verbose  
(-v in the Linux world)

```
Function Do-Something {  
    [CmdletBinding()]  
    Param($item)  
    Write-Verbose "You passed in $item"  
}
```

```
PS> Do-Something -Verbose -item "the item"
```

```
PS> Do-Something "item"
```

# Pipeline(ity)

Participate in the ability to deal with **objects**

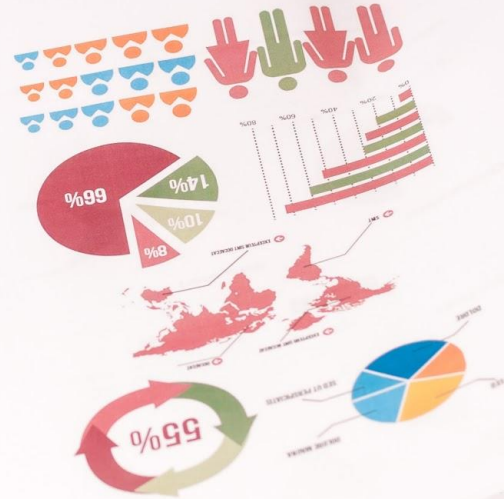
```
Function Do-Something {  
    [CmdletBinding()]  
    Param(  
        [Parameter(ValueFromPipelineByPropertyName)]  
        $Name  
    )  
    process {  
        Write-Host "You passed $Name in!"  
    }  
}  
Get-Service | Do-Something
```

# WOT IF

```
Function Remove-Something {  
    [CmdletBinding(SupportsShouldProcess)]  
    Param(  
        [Parameter(ValueFromPipelineByPropertyName)]  
        $File  
    )  
    if ($PSCmdlet.ShouldProcess($File)) {  
        Remove-Item -Path $File -Confirm:$false  
    }  
}  
touch temp  
Remove-Something temp -WhatIf
```



# Reporting



Map chart



# ConvertTo-\*

PowerShell can transform output into HTML, XML, JSON, etc.

Let's make a super special report into something that a manager can understand

```
Get-PSDrive
```

```
Get-PSDrive | ConvertTo-Html
```

```
Get-PSDrive | ConvertTo-Html | Out-File -FilePath drives.html
```

# Prettify

```
Get-PSDrive | ConvertTo-Html -Property  
Name,Used,Provider,Root,CurrentLocation | Out-File -FilePath  
PSDrives.html  
...
```

# Stylify

```
$Header = @"  
<style>  
TABLE {border-width: 1px; border-style: solid; border-color:  
black; border-collapse: collapse;}  
TD {border-width: 1px; padding: 3px; border-style: solid;  
border-color: black;}  
</style>  
"@
```

```
Get-PSDrive | ConvertTo-Html -Property  
Name,Used,Provider,Root,CurrentLocation -Head $Header | Out-File  
-FilePath PSDrives.html
```

```
$Header = @"  
<style>  
TABLE {border-width: 1px; border-style: solid; border-color:  
black; border-collapse: collapse;}  
TH {border-width: 1px; padding: 3px; border-style: solid;  
border-color: black; background-color: #6495ED;}  
TD {border-width: 1px; padding: 3px; border-style: solid;  
border-color: black;}  
</style>  
"@
```

# GUI

```
Install-Module -Name ShowUI
```

```
...
```

```
Import-Module ShowUI
```

```
Label "Hi There" -FontSize 42 -Show
```

```
Add-Type -AssemblyName System.Windows.Forms
$form = New-Object Windows.Forms.Form
$form.Size = New-Object Drawing.Size @(200,100)
$form.StartPosition = "CenterScreen"
$btn = New-Object System.Windows.Forms.Button
$btn.add_click({Get-Date|Out-Host})
$btn.Text = "Click here"
$form.Controls.Add($btn)
$drc = $form.ShowDialog()
```



# Your "short" lab then...

...

**Work on your term projects!**

**I will come by each team and I'd like a verbal report of:**

- 1) What you have remaining**
- 2) Who is doing what for the presentation**

