

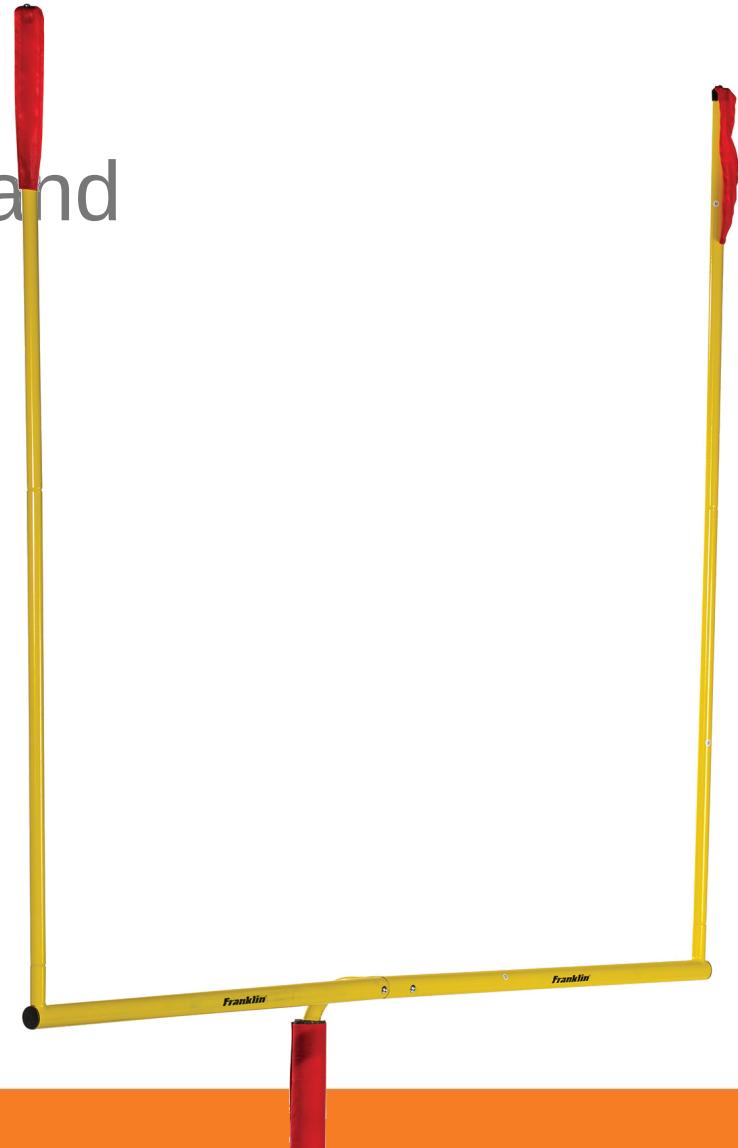
# Terminal





# Goals

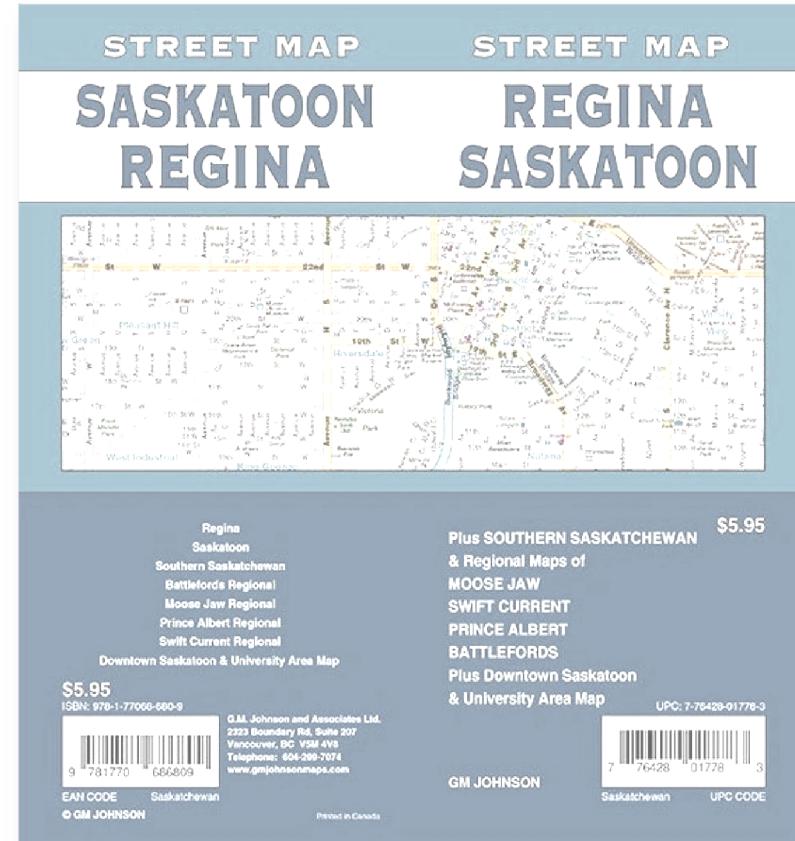
1. Define the term cmdlet
2. Explain how to get help on a specific command
3. Name an advantage of pipes





# Roadmap

1. Cmdlets
2. Pipes & Streams
3. Drives
4. Lab: Terminal Golf





Develop  
Intelligence



# Cmdlets





# Overview

- Pronounced *command-let*
- Powershell-native
- Take the form **Verb-Noun**

```
1 #Equivalent to dir or ls
2 Get-ChildItem
3
4 #Equivalent to del or rm
5 Remove-Item
6
7 #Equivalent to echo
8 Write-Output
```



# Tips

- Tab completion
  - Type Get -Com then tab
  - Works for parameters too
- Ctrl + space for a menu
- Up arrow to get previous commands
- clear aka Clear-Host to clear the console

```
1 # Get all shortcuts with this:  
2 Get-PSReadLineKeyHandler
```



# The Magic of Meta



- Cmdlets are just objects
  - Like files or services
- Interrogate them with Get-Command

```
1 # List all available
2 Get-Command
3
4 # Just process-related
5 Get-Command -Noun 'Process'
6
7 # Find out the syntax
8 Get-Command 'Stop-Process' -Syntax
```



# about\_ Parameters

- Passed by position

```
1 | Get-Command 'Get-Help'
```

- Passed by name

```
1 | Get-Command -Name 'Get-Help'
```

- Mix

```
1 | Get-Command 'Get-Help' -OnLine
```



# The Magic of Get-Help



## Commands

```
1 | Get-Help Get-ChildItem
```

## Wildcards

```
1 | Get-Help Remove-*  
2 | Get-Help about*
```

## Useful parameters

```
1 | Get-Help Get-Random -Detailed  
2 | Get-Help Get-Date -Full  
3 | Get-Help Invoke-RestMethod -Online
```



# Aliases



- Alternative version of command to
  - Save keystrokes (allegedly)
  - Feel familiar

Pwsh

Get-ChildItem

Set-Location

Write-Output

New-Item

Remove-Item

**Shorthand**

gci

sl

write

ni

ri

**Cmd.exe**

dir

chdir

echo

md

del

**Bash**

ls

cd

echo

mkdir

rm, rmdir



# Combining Commands



- Statements run sequentially

```
1 | Write-Output "Hello"; Write-Output "World"
```

- Conditional on **success**

```
1 | Set-ExecutionPolicy RemoteSigned && Write-Output "Success!"
```

- Conditional on **failure**

```
1 | Write-Output $xyz || Write-Output 'Variable "$xyz" not found'
```



# More

- Gets stuff one page at a time
- Mostly the same on:
  - Windows: more .com
  - \*Nix: /usr/bin/more
- Example:

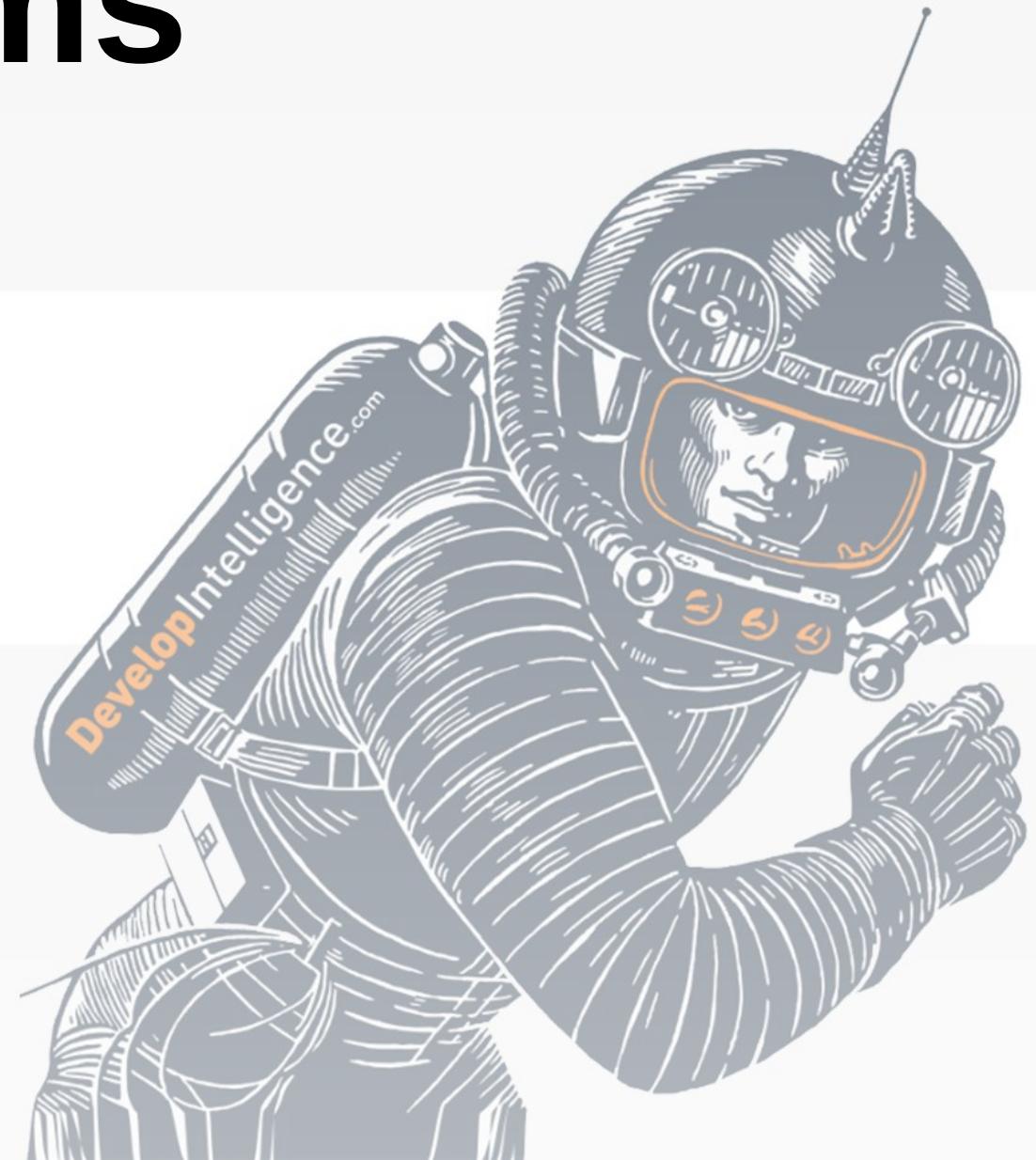
```
1 # Classic
2 Get-Command | more
3
4 # Alternative
5 Get-Command | Out-Host -Paging
```



Develop  
Intelligence



# Pipes & Streams





# Motivation

## This kind of sucks

```
1 $files = Get-ChildItem -Path *.txt
2 $longFiles = Where-Object -InputObject $files {$_.length -gt 10000}
3 $sortedFiles = Sort-Object -InputObject $longFiles -Property length
4 Format-Table -InputObject $sortedFiles -Property name, length
```



# Overview



- Series of commands connected by pipeline operators (|)
- Sends the results of the preceding command to the next command

```
1 Get-ChildItem -Path *.txt |  
2   Where-Object {$_ .length -gt 10000} |  
3     Sort-Object -Property length |  
4       Format-Table -Property name, length
```



# Multiline

```
1 # 1
2 Get-Process notepad |
3   Stop-Process
4
5 #2
6 Get-ChildItem -Path *.txt |
7   Where-Object {$_.length -gt 10000} |
8   Sort-Object -Property length |
9   Format-Table -Property name, length
```



# Filtering

- Cmdlet: Where-Object
- Alias: ?
- \$\_.is an automatic variable representing the current item in the pipeline

```
1 # Commands containing 'job'  
2 Get-Command | Where-Object {$_.Name -like '*job*'}  
3  
4 # Writeable files  
5 Get-ChildItem -File | ? -not IsReadOnly
```



# Sorting

- Cmdlet: Sort-Object

```
1 # Verbs, alphapetized
2 Get-Verb |
3     Sort-Object -Property Verb
4
5 # Biggest pigs
6 Get-Process |
7     Sort-Object -Property WS
```



# Transforming

- Cmdlet: Foreach-Object
- Alias: %
- Works like a map idiom

```
1 # Report on service status
2 Get-Service | Foreach-Object {$_.Name + " is " + $_.Status}
3
4 # Get Squares
5 1..10 | %{ $_ * $_ }
```



# Redirection

- Create a new file with >
- Append to an existing file with >>

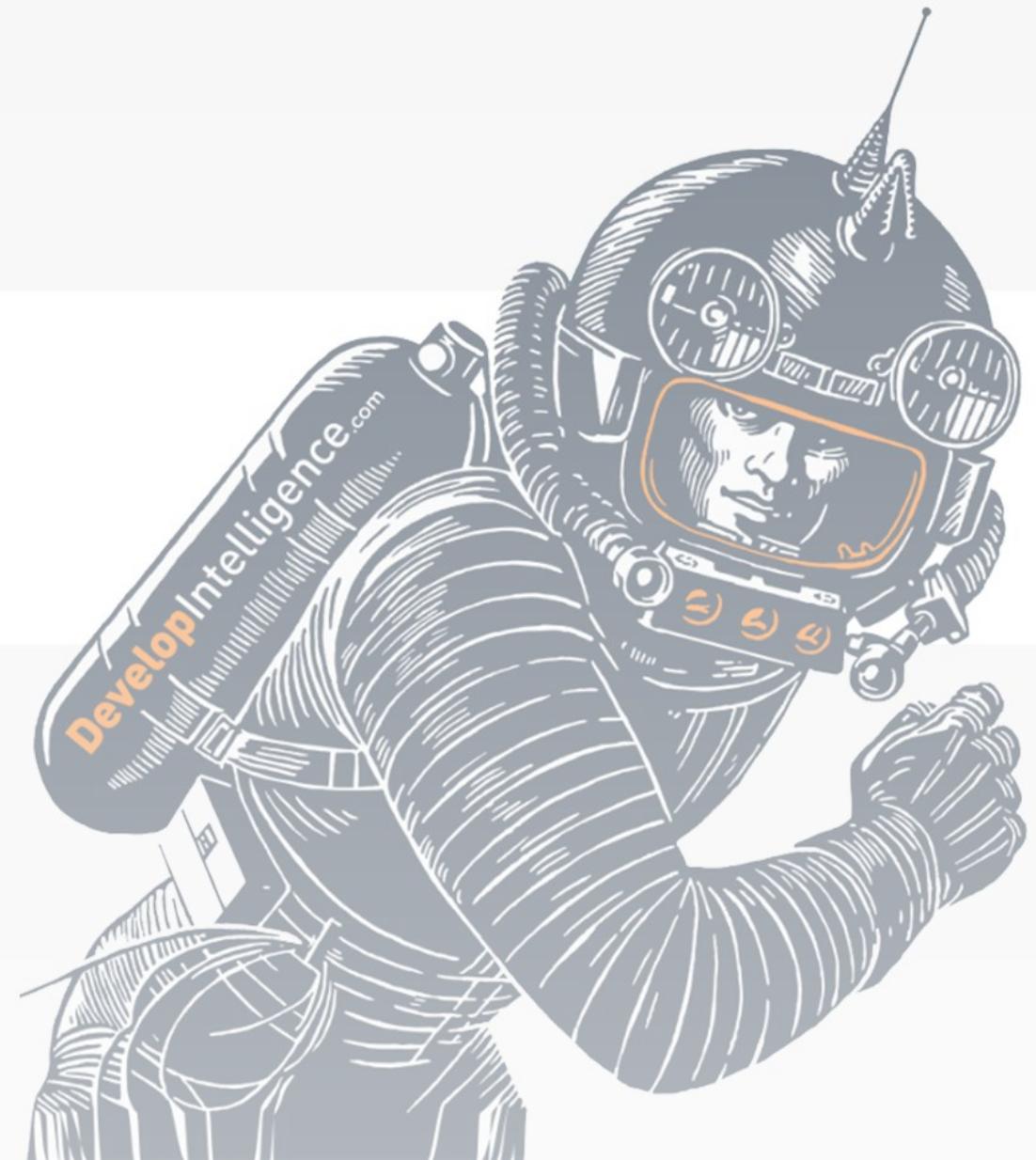
```
1 Write-Output "Services:`n" > system_state.log
2 Get-Service >> .\system_state.log
3
4 Write-Output "Processes:`n" >> system_state.log
5 Get-Process >> .\system_state.log
6
7 Get-Content .\system_state.log
```



Develop  
Intelligence



# Drives





# Overview

- The drive is central abstraction for
  - Navigation
  - C.R.U.D. operations
- Good for files, but also:
  - Registry keys
  - WMI / CIM
  - Functions & variables
- Cmdlet: [Get-PSDrive](#)



# Example: File System



```
1 Test-Path C:\Temp\  
2 New-Item -Path c:\temp -ItemType Directory  
3  
4 # Equivalent to cd  
5 Set-Location C:\temp\  
6 New-Item .\notes.txt -Value 'TODO: Remember the milk'  
7  
8 #Equivalent to cat, type  
9 Get-Content .\notes.txt
```



# Providers



- Expose hierarchical data
- Consistent paradigm
- Build-in:
  - File system
  - Registry
  - Environment
- Also:
  - Azure



# Core Commands



- Common interface for all Providers
- Examples
  - New-Item
  - Set-Location
  - Get-Item
  - Get-ChildItem
  - Remove-Item



# More Exotic Examples



## Registry

```
1 # 'CD' into a registry key
2 Set-Location 'HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\
3
4 # Create a value to fire up notepad on startup
5 Set-ItemProperty -Path . -Name 'editor' -Value 'notepad.exe'
```

## Environment Variables

```
1 Set-Location 'Env:\'
2 Get-ChildItem
```



# Provider Complaints



- Leaky abstraction
- Inconsistent implementation
  - e.g. Registry values are properties, not files
- Not always applicable
  - e.g. Environment variables aren't hierarchical



Develop  
Intelligence



# Lab: Terminal Golf





# Lab: Terminal Golf



- Read instructions: `$/labs/terminal-golf/README.md`
- Complete 9 holes
- No file editing necessary





Develop  
Intelligence





# Review

1. Define the term cmdlet
2. Explain how to get help on a specific command
3. Name an advantage of pipes

