

Powershell

コマンドレット

・ <http://technet.microsoft.com/ja-jp/library/dn169200.aspx>

- 1.get-command
- 2.get-help get-command
- 3.get-command | select-string "select"

コマンドレット	内容
Add-History	
Clear-Content	
Compare-Object	
Convert-Path	
ConvertTo-Html	
Copy-Item	
Export-Alias	
Export-Clixml	
Export-Csv	
Foreach-Object	
Format-List	
Format-Wide	
Get-Acl	
Get-Alias	
Get-AuthenticodeSignature	
Get-ChildItem	
Get-Command	
Get-Content	
Get-Culture	
Get-Date	
Get-Eventlog	
Get-ExecutionPolicy	
Get- <u>H</u> elp	
Get-History	
Get-Host	
Get-Item	

Get-ItemProperty	
Get-Location	
Get-Member	
Get-Process	
Get-PSDrive	
Get-PSProvider	
Get-Serv <u>i</u> ce	
Get-Unique	
Get-WMiObject	
Group-Object	
Import-Alias	
Import-Clixml	
Import-Csv	
Invoke-Expression	
Invoke-History	
Invoke-Item	
Measure-Command	
Measure-Object	
Move-Item	
New-Alias	
New-Item	
New-PSDrive	
New-Timespan	
Out-File	
Out-Host	
Output-Printer	
Read-Host	
Remove-Item	
Remove-PSDrive	
Rename-Item	
Restart-Serv <u>i</u> ce	
Resume-Serv <u>i</u> ce	
Select-Object	
Select-String	

Set-Alias	
Set-Content	
Set-Date	
Set-ExecutionPolicy	
Set-Location	
Set-Service	
Sort-Object	
Start-Service	
Start-Sleep	
Stop-Process	
Stop-Service	
Suspend-Service	
Tee-Object	
Test-Path	
Where-Object	返されるデータのフィルタ処理
Write-Host	
Write-Warning	

Get-Command

- ・すべての Windows PowerShell コマンドレットの一覧を返す

Get-Help

- ・PowerShell には、コマンド ヘルプと概念に関するヘルプの 2 種類のヘルプがある

■使用できるすべてのヘルプ トピック一覧を取得

Get-Help *

■トピック名を Get-Help に渡すことで特定のトピックに関するヘルプを取得できる

Get-Help Get-Member | more

Select-String

- ・文字列値の存在確認
- ・エイリアス sls

grep の代替として

- ・grep で使えるようにしておく

Set-Alias grep Select-String

Get-Alias

- ・ Windows PowerShell エイリアスのリストを返す
- ・ エイリアス gal

書式

- ・ 特定のオブジェクトについて、表示するプロパティを制御するための一連のコマンドレットが用意されています
- ・ これらすべてのコマンドレットの名前は、Format という動詞で始まります
- ・ Format 系のコマンドレットには、Format-Wide、Format-List、Format-Table、および Format-Custom があります

■ Format-Wide

- ・ オブジェクトの既定のプロパティだけが表示されます

```
PS> Get-Process -Name powershell | Format-Wide
```

■ Format-List

- ・ リスト形式で表示されます。各プロパティがその名前と共に別々の行に表示されます

```
PS> Get-Process -Name powershell | Format-List
```

■ Format-Table

```
PS> Get-Process -Name powershell | Format-Table -Property Path,Name,Id,Company
```

コマンドサンプル

- ・ 「Linux ならできるのに、だから Windows は ...」 「それ PowerShell でできるよ」
- ・ grep コマンドと PowerShell の sls (Select-String) の比較
- ・ Windows PowerShell でのパイプ処理とパイプライン
- ・ PowerShell の Get-ChildItem コマンドレットでファイル名の一覧を取得する（基本編）

■ フォルダ配下のファイル名の検索

- ・ Where-Object

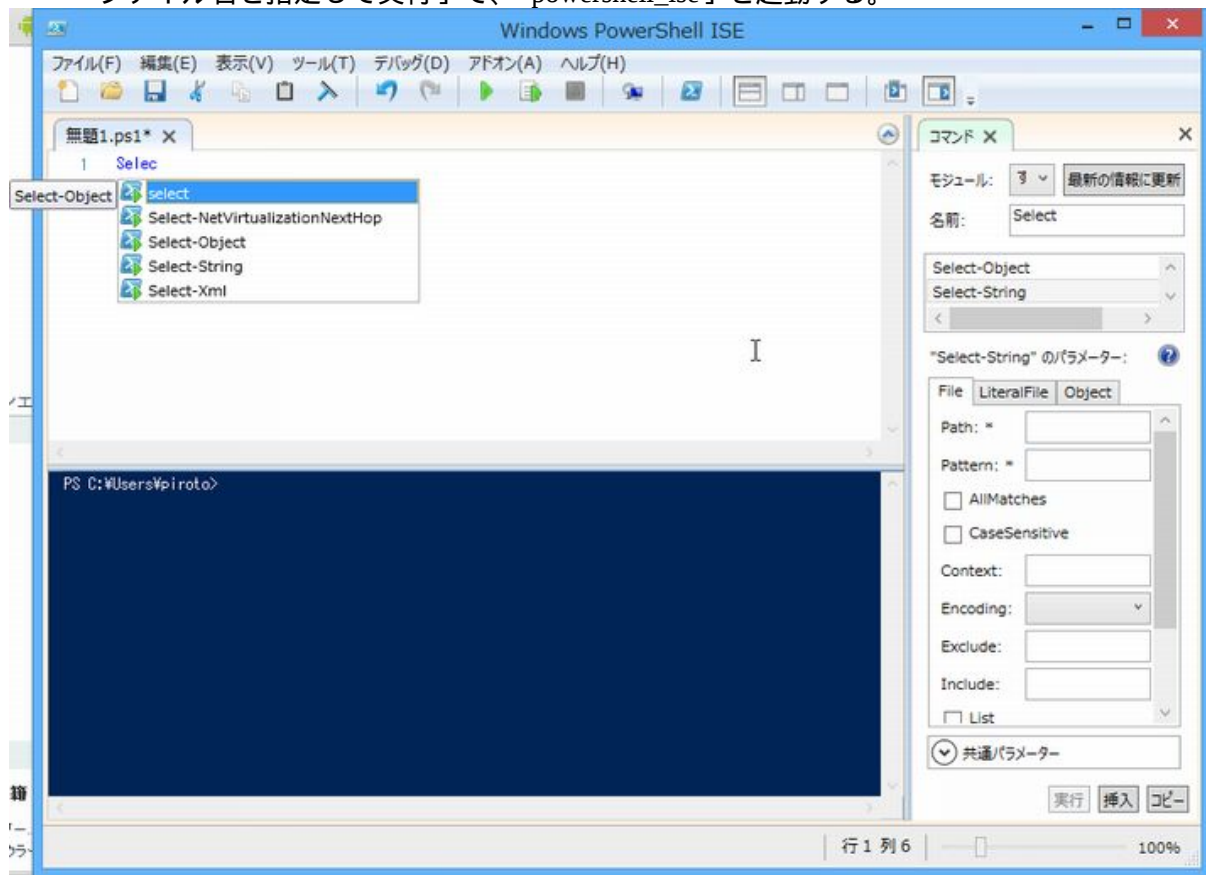
```
> get-childitem | where-object {$_.name -like "Python*"}
```

Powershell ISE

Windows8

- ・ Windows 8 のデフォルト設定では PowerShell ISE がスタート画面に表示されません。また、アプリの検索を行っても出てきません。

- ・「PowerShell ISE」には、補完、デバッグ機能などの便利な機能が付いている
- ・「ファイル名を指定して実行」で、「powershell_ise」を起動する。



Tips

スクリプトファイルを実行する

- ・ <http://www.atmarkit.co.jp/ait/articles/0709/20/news125.html>
- ・ PowerShell はカレント・フォルダを検索しない

PS > ./hoge.ps1

■ スクリプト実行ポリシーを設定

- ・ 以下のエラーが発生する

このシステムではスクリプトの実行が無効になっているため ...

PS > Set-ExecutionPolicy RemoteSigned

ポリシー	内容
Restricted	すべてのスクリプトを実行不可
AllSigned	すべてのスクリプトに証明書を要求
RemoteSigned	インターネット経由でダウンロードしたスクリプトのみ証明書を要求

Unrestricted	すべてのスクリプト実行を許可（ただしインターネット経由でダウンロードしたコードは実行確認のみあり）
--------------	---

環境変数

■ 一覧表示

```
PS> get-childitem env:
```

■ 指定表示

```
$Env: 環境変数名
```

例

```
PS> $env:path
```

■ 設定

```
PS> set-item env:HOGGE hoge
```

文字列分割

- ・ 環境変数を ";" で分割

```
PS> $Env:path -split ";"
```

バックグラウンドジョブの実行と確認

- ・ Start-Job、Get-Job

```
Start-Job -ScriptBlock{ 実行するコマンド }
```

例

```
PS C:\Programs\wildfly-8.0.0.Final\bin> start-job -ScriptBlock { .\standalone.bat }
```

Id	Name	PSJobTypeName	State	HasMoreData	Location	Command
2	Job2	BackgroundJob	Running	True	True	localhost

```
PS C:\Programs\wildfly-8.0.0.Final\bin> get-job
```

Id	Name	PSJobTypeName	State	HasMoreData	Location	Command
2	Job2	BackgroundJob	Failed	False	False	localhost

カレントディレクトリをエクスプローラーで開く

```
> explorer (get-location).path
```

コマンド情報の確認 (whereis)

```
PS C:\workspaces\vscode\reactlesson> get-command git
```

CommandType	Name	Version	Source
Application	git.exe		
Files	Git\cmd\git.exe	2.12.2.2	C:\Program