# プログラム情報

## 予約語、メタキャラクタなど

### 一般

&　Call 演算子

コマンドやプログラムのパスに特殊文字が含まれていたり、変数内に格納されたコマンド文字を実行する場合に用いる。

例

& ".\\env\\Scripts\\activate"

$command = "Get-Process"

& command

### 算術演算子

+ 　　　　2つの値を加算する

- 　　　　一方の値から他方の値を減算する

- 　　　　値を負の数値に変換する

\* 　　　　2つの値を乗算する

/ 　　　　2つの値で除算を行う

% 　　　　除算で発生した余りを返す

### 比較演算子

-eq 　　　　　等しい

-ne 　　　　　等しくない

-gt 　　　　　より大きい

-ge 　　　　　以上

-lt 　　　　　より小さい

-le 　　　　　以下

-like 　　　　ワイルドカードによる比較

-notlike　　　ワイルドカードによる比較

-match　　　　正規表現による比較

-notmatch 　　正規表現による比較

-contains 　　左側のコレクションに右側の値が含まれている 3,4,5 -contains 4 $true

-notcontains　左側のコレクションに右側の値が含まれていない 3,4,5 -contains 4 $true

### エスケープ文字

|  |  |
| --- | --- |
| エスケープ文字 | 説明 |
| `0 | null |
| `a | alert |
| `b | backspace |
| `f | form feed |
| `n | new line |
| `r | carriage return |
| `t | tab |
| `v | vertical quote |

## プログラム基本

### if

if (($a -eq $null) -And ($b -eq $null)) {

Write-Output "TRUE"

} else {

Write-Output "FALSE"

}

if (($a -eq $null) -Or ($b -eq $null)) {

Write-Output "TRUE"

} else {

Write-Output "FALSE"

}

Not

if (!($a -eq $null)) {

Write-Output "TRUE"

} else {

Write-Output "FALSE"

}

if (-Not($a -eq $null)) {

Write-Output "TRUE"

} else {

Write-Output "FALSE"

}

## 時間処理

## コマンド

Set-Location

Write-Host 出力をホストに書き込む

Write-Output オブジェクトをコンソールに表示

## ファイル系

テキストファイルに追記

Add-Content "test.txt" "hoge"

## サービス系

Start-Job

バックグラウンドで実行

Start-Job -ScriptBlock {" "}

公式：<https://learn.microsoft.com/ja-jp/powershell/module/microsoft.powershell.core/start-job?view=powershell-7.3>

Wait-Job

ジョブの終了を待つ

## ネットワーク系

### Test-NetConnection

特定ポートの疎通を確認する

Test-NetConnection -ComputerName 192.168.214.35 -Port 8008

## 環境変数

「パスを通す」例

|  |  |
| --- | --- |
| 項目 | コマンド |
| 表示 | $env:path  $env:path.split (";") |
| 追加(先頭) | $ENV:Path="追加したいパス;"+$ENV:Path  または  Set-Item Env:Path "追加したいパス;$ENV:Path" |
| 追加(末尾) | $ENV:Path+=";追加したいパス"  または  Set-Item Env:Path $Env:Path";追加したいパス" |
| 削除 | Set-Item ENV:Path $ENV:Path.Replace("削除したいパス;", "") |
| 置換 | Set-Item ENV:Path $ENV:Path.Replace("置換前のパス;", "置換後のパス;") |

※powershellは大文字小文字を区別しない

Get-ChildItem env: # 環境変数をすべて確認

echo $env:path # １つだけ確認する場合

## 管理系

### サービス管理

Start-Service –Name myservice.service # サービスの開始

Stop-Service –Name myservice.service # サービスの停止（本格的な保守作業に入る場合）

Restart-Service -Name myservice.service # サービスの再開。編集したファイルの適用。

Get-Service -Name myservice.service # サービスの状態の確認

Get-Process # 全プロセスの状態確認

Get-Process | Select-String -Pattern "myservice" # パイプを使って検索する例

Get-Process -Name myservice.service # 特定の名前のプロセスの状態確認

Stop-Process {ID} # プロセスの停止

### 文字コード

chcp # 現在の文字コードを調べる

chcp 65001 # 文字コードを指定する 65001はUTF-8

## 実行ポリシー

確認方法

powershell

get-executionpolicy # 確認

set-executionpolicy 値 # 設定（要管理者権限）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 実行ポリシー | 署名付き | 署名なし | | 備考 |
| ローカル | Web経由 |
| Restricted | × | × | × |  |
| AllSigned | ○ | × | × |  |
| RemoteSigned | ○ | ○ | × |  |
| Unrestricted | ○ | ○ | △ |  |
| Bypass | ○ | ○ | ○ |  |

例

Set-ExecutionPolicy AllSigned -Scope CurrentUser -Force #

powershell -ExecutionPolicy RemoteSigned -File D:\test.ps1 # 一時的に変更する場合

オプション

-Scope

MachinePolicy グループポリシーでそのコンピュータの全ユーザに強制

UserPolicy グループポリシーでそのコンピュータの現在のユーザに強制

Process 現在のセッションに対して適用

CurrentUser 現在のユーザに対して適用

LocalMachine そのコンピュータの全ユーザに対して適用

-Force

実行ポリシーを変更しますか? の質問を無視して強制的にポリシーの変更を実行する。

# メモ

HOSTNAME.EXE ホスト名