Шпаргалка PowerShell

Переменные

$variable = 123	Объявить переменную
$variable.GetType().Name	Узнать тип данных (Int32)
$variable.ToString()	Изменить тип данных (из числа в строку)
[string]$variable = 123	Задать тип данных
$Global:var = "global"	Глобальное объявление (действует за пределами функции)
$env:HOMEPATH / $home / ~	Переменная окружения

Массивы

$arr = @()	Объявить пустой массив (array)
$arr = "a","b","c"	Массив из строк (string)
1,2,3,4,5 / 1..5	Массив из цифр (int) / срез
$arr[0]	Получить первое значение массива
$arr[1..2]	С второго по третий индекс
$arr[-1]	Последний элемент массива
$arr[-1..-2]	Первый и второй с конца
$arr[3..2] + $arr[1..0]	Объединить два массива в один
$arr.Count	Количество элементов в массива
$arr[0].Length	Количество символов элемента
$arr[($arr.Count-1)..0]	Пересобрать массив с конца
$arr += 6	Добавить в массив новый элемент
$arr[0] += 9	Добавить (текст) или прибавить (число) к первому элементу массива

Ассоциативный массив (хэш-таблицы) и функции

$hash = @{a=1;b="test"}	Инициализировать
$hash.Keys	Список всех ключей
$hash.a = 2	Изменить значение
$hash.a / $hash["a"]	Отобразить по ключу
$hash = $hash + @{c=3}	Добавить новый ключ
$hash.Remove("c")	Удалить ключ
function test ($param) {$param}	Создать функцию
test -param "input to param"	Вызвать функцию

Справка

Get-Command g*	Поиск команды по имени
Get-Help gc	Синтаксис команды
Get-History	Вывести историю команд текущей сессии

Comments, Escape Characters, Backtick

#Comment	Комментарий
<# comment #>	Multiline comment
"A `"test`""	Escape char `
`t	Tab
`n	New line
`	Line continuation

Директории и файлы

Get-Location	Текущая директория
Set-Location	Сменить директорию
Get-Content	Вывести содержимое файла
Get-ChildItem	Содержимое папки
Test-Path	Проверить путь
Out-File	Передать в файл
Out-Null	Передать в $null
New-Item	Создать Dir/File
Get-ItemProp	Свойства
Copy-Item	Копировать
Remove-Item	Переместить
Rename-Item	Переименовать
Clear-Item	Удалить содержимое
Remove-Item	Удалить Dir/File

Управление данными

1..10 | ForEach-Object {$_+10}	Цикл потока
$x = 1; while ($x -lt 10) {$x; $x++}	Цикл while
for ($i=0; $i -lt 10; $i++) {$i}	Цикл for
foreach ($file in gci C:\) {$file.name}	Цикл foreach
$x = 5 if ($x -lt 5) {"$x меньше 5"} elseif ($x -gt 5) {"$x больше 5"} else {"$x равно 5"}	Условие

Операторы присваивания, сравнения и логические

=,+=,-=,*=,/=,%=,++,--	Операторы переменной/массива
-and / -or / -not (!)	Операторы условий: и / или / нет
-eq / -ne	Абсолютное равно / не равно
-gt / -ge	Больше / больше или равно
-lt / -le	Меньше / меньше или равно
"мама" -replace "м","п"	Замена (мама на папа)
-match,-notmatch	Приблизительное совпадение
-like,-notlike	Точное совпадение (* - wildcard)
@(1,2,3) -contains 3	Проверка присутствия в массиве
@(1,2) -notcontains 3	Проверка отсутствия в массиве

Другие операторы и регулярные выражения

-split	Разбить строку на массив: "12321" -split "2"
-join	Объединить массив в строку @(10,20) -join ":"
..	Оператор диапазона: 1..10 | % {$_ * 5}
-is,-isnot	Логический оператор выхождения
-as	Конвертировать объект в спецефический тип .NET
-f	Format strings 1..10 | foreach { "{0:N2}" -f $_ }
[ ]	Cast operator. [datetime]$birthday = "1/10/66"
$( )	Subexpression operator
@( )	Array subexpression operator
&	The call/invocation operator.

Объекты

Get-Date | Get-Member	Список свойств и методов объекта
$(Get-Date).Date / Get-Date | Select-Object Date	Получить свойство объекта по его имени
[DateTime]::UtcNow	Статическое свойство (время в формате UTC 0)
$(Get-Date).AddHours(-3)	Использовать метод (вычесть 3 часа)
$obj = [PSCustomObject]@{a=1;b=2}	Создать объект
$obj | Add-Member NoteProperty Arr @(1,2,3)	Добавить метод (новое свойство)

Filter, Sort, Group and Format (aliases for brevity)

dir C:\pub | where-object LastWriteTime -gt (Get-Date).addDays(-1)	Files in C:\pub with lastwritetime greater than yesterday
ps | where-object {$_.path -like "C:\windows\system32*" -and $_.company -notlike "Microsoft*"}	Processes where path includes system32 and company doesn't start with Microsoft
ps Explorer | select-object -Property ProcessName -ExpandProperty Modules | format-list	Get explorer processes, select processname, expand modules property array
ps | Sort-Object -Property WorkingSet | Select-Object -Last 5	Sort Processes by workingset, select last 5
"a","b","a" | Select-Object -Unique	Return only unique - returns @(a b)
Get-Service | Group-Object Status	Group services by their Status
dir | Group-Object {$_.Length -gt 100KB}	Group objects bigger/smaller than 100 KB
Get-Content C:\pcs.txt | Select-String "q-" | sls "win7"	Select strings with "q-", "win7" from pcs.txt
ps | Format-Table -Property Name, StartTime -AutoSize	Format ps output showing Name, StartTime properties, autosize the table
ps | Format-table ProcessName, @{ Label = "Total Run Time"; Expression={(Get-Date) - $_.StartTime}}	Table showing processname, custom label/expression showing run time.
Get-EventLog -Log System | Select -first 5 | Format-table -wrap	Get first 5 events in system log, wrap display
gi C:\Users | format-list -property *	Get all properties from C:\users in list format
"{0}`t{1}`n" -f $a, 5	-f operator to construct strings. {0} replaced with $a, {1} with 5 etc.

Common commands

Get-EventLog	Get-WinEvent
Get-CimInstance	Get-Date
Start-Sleep	Compare-Object
Start-Job	Get-Credential
Test-Connection	New-PSSession
Test-Path	Split-Path

Importing, Exporting and Converting

Export-CliXML	Import-CliXML
ConvertTo-XML	ConvertTo-HTML
Export-CSV	Import-CSV
ConvertTo-CSV	ConvertFrom-CSV

Automatic variables

$_, $PSItem	Current pipeline object
$Args	Script or function arguments
$Error	Errors from commands
$True,$False	Boolean value for true,false
$null	Empty
$profile	Array of profile locations

PSDrives

Alias:	Aliases in current session
Cert:	Certificate store for user
Env:	Environment variables
Function:	All functions in current session
HKLM:	Hkey Local Machine Hive
HKCU:	Hkey Current User Hive
Variable:	Variables in the current session
WSMan:	WinRM configuration / credentials
AD:	Active Directory
Set-location HKLM:	HKLM Registry hive
gci variable:	Variables in current session

Regular Expressions

\w	Any word character [a-zA-Z0-9]
\W	Any non-word character
\s	Any whitespace character
\S	Any non-whitespace character
\d \D	Any digit or non-digit
{n} {n,} {n,m}	Match n through m instances of a pattern.
More	Google .NET Regular Expressions