

PowerShell Commands >_

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
Get-ChildItem variable:	Список всех переменных

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

Директории и файлы

Get-Location	Текущая директория (pwd)
Set-Location	Сменить директорию (sl/cd)
Get-Content	Вывести содержимое файла (gc/cat)
Get-ChildItem	Содержимое папки (gci/ls)
Test-Path	Проверить путь
Out-File	Передать в файл
Out-Null	Передать в \$null
New-Item	Создай File/Directory
Get-ItemProp	Свойства
Copy-Item	Копировать (cp)
Move-Item	Переместить (mi/mv)
Rename-Item	Переименовать
Clear-Item	Удалить содержимое
Remove-Item	Удалить (ri/rm)

Управление объектами (через pipe)

Select-Object	Выборка по свойству
Format-Table	Вывод в формате таблицы
Format-List	Вывод в формате строк
Measure-Object	Подсчет элементов
Compare-Object	Сравнение
Where-Object	Фильтрация
Sort-Object	Сортировка по свойству
Group-Object	Группировка

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)

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 массива

=,+=,-=,*=,/=,%=,++,--	Операторы переменной / массива
-and / -or / -not (!)	Операторы условий: и / или / нет
-eq (-ceq) / -ne (-nce)	Логическое равно / не равно (per.)
-gt / -ge	Больше / больше или равно
-lt / -le	Меньше / меньше или равно
@(1,2,3) -contains 3	Проверка присутствия в массиве
-like / -notlike ""*Get*"	Точное равенство (* - wildcard)
-match / -notmatch	Приблизительное совпадение
"n: 77" -match "n: (7+)"	Поиск с применением групп ()
\$matches[1]	Вывести второй результат поиска
"12321" -split "2"	Разбить строку на массив
@(10,20) -join ":"	Объединить массив в строку

Регулярные выражения (Regex)

-replace	Замена (мама на папу): "мама" -replace "м","п"
\s / \n	Пробел / новая строка
\d / \D	Любая цифра / не цифра
\w / \W	Любая буква, цифра, или знак подчёркивания
\s / \S	Пробел, табуляция, перенос строки
\b / \B	Граница слова ("test tester" -replace "\btest\b")
^ / \$	Начало / конец ("she" -replace "\$","ll")
{2} / {2,4}	Найти два / два или 4 повторения ({2,} и более)
[a-яA-Я0-9]	Диапазон символов для поиска