# #PS\_1

## **Общие знание**

*Cách gán biến:*

$v1 = 555/5

$v1

$v1\*3

PowerShell использует новое понятие «командлеты» (пишется «cmdlet»).

PowerShell sử dụng khái niệm mới về "cmdlets" (đánh vần là "cmdlet").

Командлет – это простой инструмент командной строки, интегрированный в оболочку и выполняющий единственную функцию..

Cmdlet là một công cụ dòng lệnh đơn giản được tích hợp vào hệ vỏ và thực hiện một chức năng duy nhất ..

Командлеты можно использовать поодиночке, но их мощь становится более очевидной при использовании комбинаций (конвейер) командлетов для выполнения сложных задач.

Cmdlets có thể được sử dụng một mình, nhưng sức mạnh của chúng trở nên rõ ràng hơn khi bạn sử dụng kết hợp (đường ống) của lệnh ghép ngắn để thực hiện các tác vụ phức tạp.

Типы команд:

|  |  |
| --- | --- |
| * Командлеты * Функции * Сценарии * Исполняемые файлы | • Máy tính xách tay  • Chức năng  • Kịch bản  • Tập tin thực thi |

Общий вид командлеты:

**имя\_командлеты [список\_параметров]**

Командлет без параметров:

**имя\_командлеты**

Командлет с одним параметром (tham số):

**имя\_командлеты параметр**

* **имя\_командлеты => Глагол-Существительное**
  + **Глагол** определяет действие, которое надо выполнить: **Get, Set, Stop, Clear, Sort**
  + **Существительное** определяет объект, над которым, надо выполнить действие: **Process, Help, Service, Host**
  + Пример:

**Get-Process, Get-Help, Stop-Service, Clear-Host**

* **имя\_командлета [список\_параметров]**

В **список\_параметров** могут входить следующие типы параметров:

* + переключатель; (công tắc)
  + параметр со списком значений; (tham số với danh sách các giá trị;
  + значения без указания параметра (giá trị mà không chỉ định một tham số)

Перед именем **параметра** должен стоять знак минуса (-) :

**–имя\_переключателя**

**–имя\_параметра список\_значений**

Перед **значением** **параметра** **НЕТ** знака минуса:

**значение\_без\_параметра**

Параметры отделяются друг от друга группой пробелов (один или более):

Các tham số được phân tách với nhau bằng một nhóm khoảng trắng (một hoặc nhiều):

**–имя\_переключателя**

Пример:

**Dir -Recurse -Filter n\*.exe c:\Windows**

|  |  |
| --- | --- |
| * Командлеты «Get» — извлекают данные; * Командлеты «Set» — создают или изменяют данные; * Командлеты «Format» — форматируют данные; * Командлеты «Out» — пересылают выводимые данные в указанное место. | * Truy xuất dữ liệu; * Tạo hoặc sửa đổi dữ liệu; * Định dạng dữ liệu; * Gửi đầu ra đến vị trí đã chỉ định. |

Get-Help \*

Get-Help командлет –Detailed

Например:

Get-Help Sort-Object –Detailed

Get-Help Get-Help –Detailed

Командлет –?

Get-Help командлет –Examples

Get-Help командлет –Full

## **Алиасы - Bí danh**

**Первый тип** – для совместимости имен с разными интерфейсами

dir, ls, Get-ChildItem

Loại đầu tiên là để tương thích tên với các giao diện khác nhau

**Второй тип** (стандартные псевдонимы) – для быстрого ввода команд: Get => g, Set => s, Location => L тогда

Loại thứ hai (bí danh tiêu chuẩn) là để nhập nhanh các lệnh: Get => g, Set => s, Location => L:

Get-Location => gl,

Set-Location => sl

Командлет просмотра псевдонимов:

Get-Alias или gal

Возможные параметры Get-Alias:

* –Definition имя-командлеты
* Пример:

Get-Alias –Definition Get-ChildItem

* –Name имя-алиаса
* Пример:

Get-Alias –Name g\*

## **Определение новых псевдонимов:**

New-Alias –Name имя-алиаса –Value имя-командлеты

Пример:

New-Alias –Name we –Value Get-ChildItem

**Удаление псевдонимов:**

Remove-Item Alias:имя-алиаса

Пример:

Remove-Item Alias:we

Узнать текущее положение в файловой системе: Set-Location

Изменить текущее положение в файловой системе: Get-Location

# #PS\_2

* Xem nội dung thư mục hoặc PS-Driver:

Get-ChildItem hoặc dir hoặc ls hoặc gci

* Thông số:
* Filter ← tham số với đối số
* Recurse ← Tùy chọn chuyển đổi
* Force ← Xem các tập tin ẩn
* Xem nội dung của thư mục hiện thời trên đĩa đang hoạt động:

Get-ChildItem

* Xem nội dung của ổ đĩa trong PowerShell nơi đặt các từ đồng nghĩa cmdlet:

dir Alias:

* Xem nội dung của ổ đĩa trong PowerShell nơi đặt các biến môi trường Windows và các giá trị của chúng:

dir Env:

* Tham số với Bộ lọc đối số cho phép chỉ hiển thị những tập tin thư mục thỏa mãn bộ lọc:

ex: dir C:\”Program Files” −Filter m\*

* Tùy chọn chuyển đổi Recurse cho phép bạn hiển thị nội dung của thư mục hiện tại và tất cả các thư mục con của nó:

ex: dir C:\”Program Files” −Recurse

**🗹 Задание:**

* + Вывести на экран каталог дисков: C: Alias: Cert: Env: Function:.
  + Перейти на диск C: в каталог Program Files.
  + Найти, где расположена программа Word (WinWord.exe). Использовать командлетdir с маской \*word\*.exe и параметром−Recurse

**Giải:**

dir c:

dir Alias:

dir Cert:

dir c:\'Program Files' -Recurse -Filter \*word\*.exe

## **CÁC THAO THÁC VỚI FILE / FOLDER:**

### Создание каталога:

New-Item или ni

Параметры:

|  |  |
| --- | --- |
| -Name | имя создаваемого каталога |
| -Type Directory | тип объекта - каталог |
| -Path | родительский каталог, где будет создаваться новый каталог |

Пример:

ni -Name Новый -Type Directory

### Создание файла:

New-Item или ni

Параметры:

|  |  |
| --- | --- |
| -Name | имя создаваемого файла |
| -Type File | тип объекта – файл |
| -Path | родительский каталог, где будет создаваться новый файл |
| -Force | если файла уже существует, то он будет перезаписан |
| -Value | указывается значение, которое будет записано в создаваемый файл |

Пример:

ni -Name NewF -Type File -Value “Новый файл”

### Просмотр содержимого файла:

Get-Content или type или gc или cat

Параметры:

|  |  |
| --- | --- |
| -Path | путь, который включает и имя просматриваемого файла |
| -Encoding | кодировка файла: Unicode, Byte, BigEndianUnicode, UTF8, UTF7, UTF32, Ascii ; |
| -TotalCount | указывает количество читаемых строк файла. |

### Копирование файла:

Copy-Item или copy или cp или cpi

Параметры:

|  |  |
| --- | --- |
| -Path | путь, который включает и имя просматриваемого файла |
| -Destination | место, куда надо скопировать файл; |
| -PassThru | формирует сообщение о выполненных операциях копирования. (tạo một thông báo về các hoạt động sao chép hoàn thành.) |

Пример:

copy -Path NewF -Destination Новый -PassThru

### Переименование файла:

Rename-Item или ren или rni

Параметры:

|  |  |
| --- | --- |
| -Path | путь, который включает и имя переименоваемого файла; |
| -Destination | новое имя файла; |
| -PassThru | формирует сообщение о выполненных операциях переименования. |

Пример:

ren Ndf -NewName helou.txt -PassThru

### Перемещение файла:

Move-Item или move или mi или mv

Параметры:

|  |  |
| --- | --- |
| -Path | путь, который включает и имя перемещаемого файла; |
| -Destination | место, куда будет перемещаться файл; |
| -PassThru | формирует сообщение о выполненных операциях переименования. |

Пример:

move hai2\helou.txt -PassThru -Destination .

### Удаление файла или каталога:

ReMove-Item или del или erase или ri или rd или rm или rmdir

Параметры:

|  |  |
| --- | --- |
| -Path | путь, который включает и имя удаляемого файла (или каталога) |
| -WhatIf | не удаляет, а только показывает что будет удалено. |
| -Recurse | удаляет содержимое каталога и всех его подкаталогов |
| -Exclude | какие объекты не удаляют; |
| -Filter | какие объекты удаляются; |

Создайте каталог, в имени которого присутствует текущая дата:

Ni -Type Directory -Name “Пример $(Get-Date -Uformat %y%m%d)”

### **PS-DRIVER**

#### Провайдеры и диски:

(Nhà cung cấp và ổ đĩa:)

**Kiến thức tổng quát:**

Диски в PowerShell – это различные хранилища данных.

Đĩa trong PowerShell là các kho lưu trữ dữ liệu khác nhau.

Хранилища данных связаны с конкретным Провайдером (Поставщик). И структура данных и операции над ними зависит от Провайдера.

Kho dữ liệu được liên kết với một Nhà cung cấp (Nhà cung cấp) cụ thể. Và cấu trúc dữ liệu và hoạt động trên chúng phụ thuộc vào Nhà cung cấp.

Провайдеры (Поставщики) PowerShell — это программы, которые предоставляют пользователям PowerShell доступ к данным, содержащимся в специализированных хранилищах.

Nhà cung cấp PowerShell là các chương trình cung cấp cho người dùng PowerShell quyền truy cập vào dữ liệu có trong kho lưu trữ chuyên dụng.

Провайдеры позволяют просматривать эти данные и управлять ими.

Các nhà cung cấp cho phép bạn xem và quản lý dữ liệu này.

Такие данные размещаются аналогично данным в файловой системе на диске.

Dữ liệu đó được đặt tương tự như dữ liệu trong hệ thống tệp trên đĩa.

Danh sách các Nhà cung cấp có thể thu được bằng cmdlet:

Get-PSProvider

|  |  |
| --- | --- |
| **Поставщик (Provider)** | **Описание** |
| [**Alias**](http://winintro.ru/windowspowershellhelp.ru/html/145686e2-7421-4fef-ba58-165a5704e6d8.htm) | Предоставляет доступ к псевдонимам Windows PowerShell и их значениям.  (Cung cấp quyền truy cập vào bí danh Windows PowerShell và ý nghĩa của chúng.) |
| [**Certificate**](http://winintro.ru/windowspowershellhelp.ru/html/379805bb-ea22-4529-825d-7973be92a3fe.htm) | Предоставляет доступ только для чтения к хранилищам сертификатов X509 и сертификатам.  (Cung cấp quyền truy cập chỉ đọc vào các cửa hàng chứng chỉ và chứng chỉ X509.) |
| [**Environment**](http://winintro.ru/windowspowershellhelp.ru/html/b58db096-d6db-4409-9e19-14b1edae988c.htm) | Предоставляет доступ к переменным среды Windows.  (Cung cấp quyền truy cập vào các biến môi trường Windows.) |
| [**FileSystem**](http://winintro.ru/windowspowershellhelp.ru/html/a7665c41-1092-4617-8715-797bbb70ac3f.htm) | Предоставляет доступ к файлам и каталогам.  (Cung cấp quyền truy cập vào các tập tin và thư mục.) |
| [**Function**](http://winintro.ru/windowspowershellhelp.ru/html/73ceb9b2-7c7a-4268-b43b-ee2f429dbbcc.htm) | Предоставляет доступ к функциям, определенным в Windows PowerShell.  (Cung cấp quyền truy cập vào các tính năng được xác định trong Windows PowerShell.) |
| [**Registry**](http://winintro.ru/windowspowershellhelp.ru/html/fe96fa76-3919-44be-b396-adb05cb036f5.htm) | Предоставляет доступ к разделам системного реестра и их значениям.  (Cung cấp quyền truy cập vào các khóa registry và các giá trị của chúng.) |
| [**Variable**](http://winintro.ru/windowspowershellhelp.ru/html/f3595a74-0a49-43e6-88d1-7eba58dcc1e4.htm) | Предоставляет доступ к переменным Windows PowerShell и их значениям.  (Cung cấp quyền truy cập vào các biến Windows PowerShell và các giá trị của chúng.) |
| [**WSMan**](http://winintro.ru/windowspowershellhelp.ru/html/66fe1241-e08f-49ca-832f-a84c33ca8735.htm) | Предоставляет доступ к сведениям о конфигурации WSMan.  (Cung cấp quyền truy cập vào thông tin cấu hình WSMan.) |

Формат выводимой информации о содержимом каталога зависит от провайдера, которому соответствует диск.

Định dạng của thông tin đầu ra về nội dung của thư mục phụ thuộc vào nhà cung cấp mà đĩa tương ứng.

Эту информацию можно получить используя командлету Get-Member в конвейере:

Thông tin này có thể được lấy bằng lệnh ghép ngắn Get-Member trong dãy lệnh:

dir c: | Get-Member

dir Alias: | Get-Member

dir Cert: | Get-Member

Виртуальные диски PowerShell (доступны только из оболочки PowerShell) (Đĩa ảo PowerShell (chỉ có sẵn từ PowerShell))

Список дисков доступных в сеансе: (Danh sách các ổ đĩa có sẵn trong phiên:)

Get-PSDrive

Get-PSDrive или gdr

Параметры:

1. имя диска
2. -PSProvider FileSystem

* **Выполните операции: (Пример)**

Get-PSDrive C

Get-PSDrive -PSProvider FileSystem

#### Создание нового диска (на время сеанса PowerShell):

Tạo một ổ đĩa mới (trong phiên PowerShell)

New-PSDrive или ndr или mount

Параметры:

|  |  |
| --- | --- |
| -Name | имя создаваемого диска |
| -Root | каталог, на который указывает виртуальный диск |
| -PSProvider FileSystem |  |

**🗹 Задание:**

1. Создайте каталог с именем PowerShell. (New-Item –Type Directory)
2. В каталоге PowerShell создайте несколько текстовых файлов, в каждый поместив по несколько строк текста. (New-Item –Type File)
3. Создайте виртуальный диск PS, который указывает на каталог PowerShell. (mount)
4. Вывести на экран список всех дисков и содержимое каталога PowerShell. (Get-PSDrive, dir)

### **Перенаправление вывода**

Chuyển hướng đầu ra

Перенаправление информации, которая должна быть выведена на экран, в создаваемый заново текстовый файл:

Chuyển hướng thông tin sẽ được hiển thị trên màn hình sang tệp văn bản mới được tạo:

**командлета > файл**

Дозапись информации в конец текстового файла:

Ghi thông tin ở cuối tệp văn bản:

**командлета >> файл**

**🗹 Задание:**

1. Создать текстовый файл на диске PS с расширением TXT. В этот файл вывести содержимое корневого каталога диска C: . (dir, >)
2. Записать в конец созданного файла список дисков. (Get-PSDrive, >>)
3. Вывести на экран содержимое созданного файла. (Type)

Перенаправление информации с использованием командлеты Out-File:

Chuyển hướng thông tin bằng lệnh ghép ngắn Out-File:

командлета |Out-File файл

Дозапись информации в конец текстового файла:

Ghi thông tin ở cuối tệp văn bản:

командлета |Out-File файл -Append

### **Подавление вывода:**

Loại bỏ đầu ra:

Out-Null

Подавление вывода информации по результатам выполнения командлеты.

Việc ngăn chặn đầu ra thông tin dựa trên kết quả của lệnh ghép ngắn.

# #PS\_3

## **Конвейер командлет**

Dãy lệnh cmdlet (pipelines)

Конвейер командлет, это объединение нескольких командлет путем последовательного перенаправления выходного потока одной командлеты во входной поток другой.

Một dãy lệnh cmdlet là sự kết hợp của một số lệnh ghép ngắn bằng cách chuyển tiếp tuần tự luồng đầu ra của một lệnh ghép ngắn sang luồng đầu vào của luồng khác.

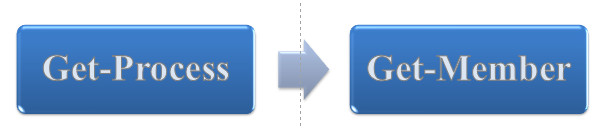
Конвейер командлет, обеспечивает комбинированное действие командлетов за счет передачи результата выполнения первой командлеты на вход второй командлеты.

Dãy lệnh cmdlet cung cấp hành động kết hợp của các lệnh ghép ngắn bằng cách truyền kết quả của lệnh ghép đầu tiên đến đầu vào của lệnh ghép ngắn thứ hai.

Конвейер командлет, записывается как последовательность командлет, разделенных символом «вертикальная черта».

Dãy lệnh cmdlet được viết dưới dạng một chuỗi các lệnh ghép được phân tách bằng một thanh dọc.

Get-Process | Get-Member



Информация между командлетами конвейера, передается в виде объектов.

Thông tin giữa các dãy lệnh được truyền dưới dạng đối tượng.

## **Объекты**

**Объект** - это совокупность данных (свойства объекта - Property) и способов работы с этими данными (методы объекта - Method).

Một đối tượng là sự kết hợp của dữ liệu (thuộc tính đối tượng - Thuộc tính) và phương thức làm việc với dữ liệu này (phương thức đối tượng - Phương thức).

dir | Get-Member

Информацию о **структуре объекта** можно получить посредством командлеты:

Thông tin về cấu trúc của một đối tượng có thể thu được thông qua lệnh:

Get-Member

Но на вход этой командлете **обязательно должен поступить объект**. Это можно выполнить через конвейер:

Nhưng đối tượng phải được nhận ở đầu vào của lệnh ghép ngắn này. Điều này có thể được thực hiện thông qua các dãy:

Get-Process | Get-Member

Можно вывести только свойства объекта (Property) или только методы объекта (Method):

Bạn chỉ có thể hiển thị các thuộc tính của một đối tượng (Thuộc tính) hoặc chỉ các phương thức của một đối tượng (Phương thức):

Пример:

dir | Get-Member -MemberType Property

dir | Get-Member -MemberType Method

**🗹 Задание:**

1. Вывести на экран все свойства объекта (Property) который, описывает процессы, выполняемые на компьютере (Get-Process).

Get-Process | Get-Member -MemberType Property

1. Вывести на экран все методы (Method) работы с процессами.

Get-Process | Get-Member -MemberType method

1. Вывести на экран свойства процессов и методы работы с ними.

Get-Process | Get-Member -MemberType Property, method

## Выбор свойств объекта

Lựa chọn thuộc tính đối tượng

Вывод только указанных свойств объекта:

Chỉ xuất ra các thuộc tính được chỉ định của đối tượng:

Select-Object список\_свойств\_через\_запятую

Сам объект поступает на командлету через конвейер.

Đối tượng tự vào cmdlet thông qua dãy lệnh.

Пример:

dir | Select-Object FullName, LastWriteTime

**🗹 Задание:**

1. Найти директорию, где имеется достаточно файлов. (Find the directory where there are enough files)
2. Вывести содержимое найденной директории. Список выводимых свойств: CreationTime, Name, Mode, Length.

## Упорядочивание элементов

Sắp xếp các thành phần

Упорядочивание (сортировку) элементов конвейера:

Đặt hàng (sắp xếp) các yếu tố dãy lệnh:

Sort-Object список\_свойств\_через\_запятую

Сортировка элементов конвейера в обратном порядке:

Sắp xếp các thành phần dãy lệnh theo thứ tự ngược lại:

Sort-Object список\_свойств -Descending

## Упорядочивание информации

Sắp xếp các thông tin

Sort-Object или sort

Пример:

dir | Sort-Object Length

dir | Sort-Object Name -Descending

Ограничить вывод количеством первых или последних записей:

Giới hạn đầu ra cho số lượng mục đầu tiên hoặc cuối cùng:

Select-Object -First количество\_первых

Select-Object -Last количество\_последних

## Объекты

Сохранить список всех программных продуктов, установленных на компьютере, в **переменную** $pr\_1:

$pr\_1 = Get-WmiObject win32\_product

Посмотреть полученный список:

$pr\_1

**🗹 Задание:**

1. Вывести список, установленного на компьютере ПО. Список выводимых свойств: Name, Vendor, InstallDate.

Get-WmiObject win32\_product | Select-Object Name, Vendor, InstallDate

1. Упорядочить выводимую информацию по дате инсталляции ПО (InstallDate) по убыванию (параметр -Descending).

Get-WmiObject win32\_product | Select-Object Name, Vendor, InstallDate | Sort-Object InstallDate -Descending

## Информации о процессах

Get-Process или ps

Пример:

Get-Process

ps | Get-Member

**🗹 Задание:**

1. Вывести список процессов (Get-Process).
2. Посмотреть свойства, описывающие процессы (Get-Member )
3. Вывести только имя процесса, идентификатор и процессное время.

gps | Select-Object Name, TotalProcessorTime, id

1. Полученный список процессов упорядочить по убыванию процессного времени.

gps | Select-Object Name, TotalProcessorTime, id | Sort-Object TotalProcessorTime -Descending | Select-Object -first 30

## Использование переменных

В переменные используются для хранения информации. Временно или постоянно (переменные Windows и PowerShell).

Các biến được sử dụng để lưu trữ thông tin. Tạm thời hoặc vĩnh viễn (biến Windows và PowerShell).

Имя переменной начинается со знака $ (доллар), далее любой набор букв, цифр и знака подчеркивания ‘\_’.

Tên biến bắt đầu bằng ký hiệu $ (đô la), sau đó là bất kỳ tập hợp chữ cái, số và dấu gạch dưới ‘\_.

Переменные PowerShell, временные и постоянные, хранятся на диске Variable.

Các biến PowerShell, tạm thời và vĩnh viễn, được lưu trữ trên đĩa **Biến**.

Присвоить значения переменной можно операцией = (знак «равно»):

$w\_1 = “Строка символов”

Вывод значения переменной на экран:

$w\_1

Вывести тип переменной на экран :

$w\_1 | Get-Member

или через метод GetType():

$w\_1.GetType().Name

**🗹 Задание:**

1. Занести произвольную текстовую строку в переменную, имя которой начинается с символа V.

gv v\*

1. Создать директорию. (New-Item)

ni -Name blah -type directory

1. Вывести в файл (в созданной директории) список всех переменных, начинающихся с символа V. (dir с диска Variable:)

ni -name var.txt -type file

gv v\* > var.txt

1. Вывести на экран содержимое созданного файла. (type)

gc var.txt

Справа от знак «равно» (=) может стоять выражение (арифметическое или строковое) или командлета (конвейер командлет).

Ở bên phải dấu bằng (=) có thể là biểu thức (số học hoặc chuỗi) hoặc lệnh ghép ngắn (đường ống cmdlet).

$w\_1 = 150 + 401

$w\_1 = “Строка символов”

$w\_1 = ( 2 -eq 2 )

$w\_1 = dir | Select-Object -First 5

**🗹 Задание:**

Для следующих операции занести в файл:

$v\_int = 150 + 401

$v\_str = “Строка символов”

$v\_log = ( 2 -eq 2 )

$v\_obj = dir c:\ | Select-Object -First 5

1. Результат каждой операции.
2. После результата каждой операции записать тип результата через GetType().Name, а потом через Get-Member.
3. Вывести на экран содержимое созданного файла.

При задании строки текста могут использоваться одинарные (‘ ’) или двойные (“ ”) кавычки.

Khi chỉ định một dòng văn bản, có thể sử dụng dấu ngoặc kép đơn hoặc dấu ngoặc kép.

При одинарных кавычках (‘ ’), текст остается без изменений. А при двойных (“ ”) – вместо переменных, расположенных в тексте, подставляются их значения.

Với dấu ngoặc đơn, văn bản vẫn không thay đổi. Và với ngoặc kép - thay vì các biến nằm trong văn bản, các giá trị của chúng được thay thế.

Пример Использование одинарных (‘ ’) или двойных (“ ”) кавычек:

|  |  |
| --- | --- |
| IN | OUT |
| $w\_1 = “Первые” |  |
| $w\_2 = 100 |  |
| ‘$w\_1 $w\_2 !’ | $v\_int $v\_log |
| “$w\_1 $w\_2 ???” | 551 True |

**🞎 Задание:**

1. В первую переменную поместить в одинарных кавычках (‘’) строку, в которой помимо текста присутствуют две переменные.

Trong biến đầu tiên, đặt các dấu ngoặc đơn (‘)) một chuỗi trong đó ngoài văn bản còn có hai biến.

1. Присвоить второй переменной числовое значение, а третей – символьное значение.

Gán một giá trị số cho biến thứ hai và giá trị tượng trưng cho biến thứ ba.

1. Вывести на экран содержимое первой переменной.

Hiển thị nội dung của biến đầu tiên.

1. Повторить п.1 с двойными кавычками (“”) и п.3.

Lặp lại bước 1 với dấu ngoặc kép (Giành) và bước 3.

### **ОПЕРАЦИИ НАД СТРОКОВЫМИ ЗНАЧЕНИЯМИ**

|  |  |  |  |
| --- | --- | --- | --- |
| **Операция** | **Описание** | **Пример** | **Результат** |
| **+** | **Объединение строк** | **$w1 = ‘a’+’z’** | **‘az’** |
| **\*** | **Повторение строки** | **‘a’\*3** | **‘aaa’** |
| **+=** | **Добавление к переменной** | **$w1 = ‘az ’**  **$w1 += ‘qw’** | **‘az qw’** |
| **\*=** | **Повторение символьной переменной** | **$w1 = ‘az ’**  **$w1 \*= 3** | **‘az az az ’** |

**🗹 Задание:**

1. Поместить строку из 4 символов в переменную $var\_1.

Đặt một chuỗi gồm 4 ký tự.

1. Повторить в переменной эту строку 3 раза.
2. Добавить в начало этой строки слово ‘Старт’, а в конец – слово ‘Финиш’.
3. Вывести на экран содержимое созданной переменной.

### **ОПЕРАЦИИ СРАВНЕНИЯ:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Операция** | **Описание** | **Пример** | **Результат** |
| **-eq** | **Равно** | **3 -eq 3** | **True** |
| **-ne** | **Не равно** | **3 -ne 3** | **False** |
| **-lt** | **Меньше** | **1 -lt 3** | **True** |
| **-le** | **Меньше или равно** | **4 -le 3** | **False** |
| **-gt** | **Больше** | **4 -gt 3** | **True** |
| **-ge** | **Больше или равно** | **1 -ge 3** | **False** |

**🗹 Задание:**

1. Поместить в 4 переменные различные числовые значения.

(Đặt vào 4 biến số giá trị số khác nhau)

1. Показать все возможные результаты выполнения всех операций сравнения.

# #PS\_4

## **ФУНКЦИИ**

Блок кода на языке PowerShell, имеющий название и находящийся в памяти до завершения текущего сеанса командной оболочки

Một khối mã PowerShell được đặt tên và lưu giữ trong bộ nhớ cho đến khi kết thúc phiên shell hiện tại

Анализ синтаксиса функции производится только один раз, при ее объявлении (при повторном запуске функции подобный анализ не производится)

Việc phân tích cú pháp của hàm chỉ được thực hiện một lần, khi nó được khai báo (phân tích như vậy không được thực hiện khi chức năng được khởi động lại)

Функция – набор команд, имеющее название и находящееся в памяти до завершения текущего сеанса.

Hàm - một tập hợp các lệnh có tên và nằm trong bộ nhớ cho đến khi kết thúc phiên hiện tại.

function имя\_функции

{

тело\_функции

}

Вызов функции:

имя\_функции

**Пример**:

*Написать функцию, которая приветсвует пользователя.*

function Hello{

“Добрый день!”

}

Вызов функции:

Hello

### Оператор ввода информации:

Nhà điều hành nhập thông tin:

$переменная = Read-Host “сообщение”

На консоль выводится сообщение, а введенное значение помещается в $переменная .

Một thông báo được hiển thị trên bàn điều khiển và giá trị đã nhập được đặt trong biến $.

**Пример**:

$a\_1 = Read-Host “Введите строку”

$a\_1

Вывод списка функции:

dir Function:

Вывод содержимого функции:

type Function:имя\_функции

Пример:

type Function:Hello

### **Использование переменных**

Показать пример для символьного сравнения и числового:

$str1 = '20'

$str2 = '-20'

$str1 -gt $str2

Потому, что:

’20’ –gt ‘-20’ => False

Но если выполнить перевод из символьного в числовой формат:

[int]$int1 = $str1

[int]$int2 = $str2

$int1 –gt $int2

Потому, что:

20 -gt -20 => True

Имеется операция «сравнения с шаблоном»:

-like

Пример:

$w1 = ‘\*x\*’

‘zxz’ -like $w1 истина

‘zzz’ -like $w1 ложно