1. Привести пример конвертации 1 урок 109 строка.

Конвертация вроде проходит, но когда смотришь тип данных, он не изменился.

[Byte]$bytevar = 3; [Double]::ToDouble(); $bytevar.GetType()

[System.int32]$a = 32; $a.GetType();$a.tostring().GetType() Урок 4 строка 120

2. 1 урок 227 строка. Как поведёт себя скрипт, если в одном из if будет получено значение false. Скрипт перестанет выполнять следующие вложенные if и перейдёт ветвлению else?

# выполняется блок кода если логическая операция вернула TRUE

if ($x -lt $y)

{

# ветвления могут быть вложенные, например:

if(3 -gt 2)

{

if(3 -gt 2)

{

}

}

}

else # выполняется блок кода если логическая операция вернула FALSE

{

}

1. 3 урок строка 177. Когда мы передаём новые аргументы по значению в функцию, то создаётся их копия с тем значением, которое им было присвоено, а когда добавляем аргумент ссылочного типа со значение в функцию, то он перезаписывается уже существующим ссылочным значением в функции. Почему?

function do-test([ref]$var){ # в данном случае создается переменная, содержащая ту же ссылку что и внешняя

$var.value = 15 # переопределение переменной по ссылке

}

$var1 = [ref]3

do-test $var1

$var1 # = 15! # значение передавалось по ссылке, соответственно $var1 приняла значение = 15.

1. 3 урок строка 201 и до конца. Необходимо закрепить

<#

# ПАРАМЕТРЫ АРГУМЕНТОВ!

Урок 4: Функции часть 2

############

параметры аргументов ------------------**немного рассмотрели в уроке 8 в конце**

############

синтаксис бегин процесс энд

function [<scope:>]<name>

{

param([type]$parameter1 [,[type]$parameter2])

dynamicparam {<statement list>}

begin {<statement list>}

process {<statement list>}

end {<statement list>}

}

1. 4 урок

######

# есть спец коммандлеты для работы с массивами:

# Format-Table - приводит массив к таблице

# Sort-object - сортирует массив по значению

# select-object - выбирает значения из массива

### слайсы массива (срезы)

### хештаблицы

1. Урок 5 классы плохо усвоился. Когда необходимо применять пользовательский ссылочный тип данных class? Только класс имеет явный и неявный конструктор? «Конструктор — это такая функция, которая вызывается каждый раз при создании объекта этого класса»

class connector{

[System.String]$name # свойство (имеет тип и имя), является членом класса

}

В создании объекта класса обязательно должно использоваться одно из свойств типа данных? Например

[string].? (name…..)

Урок 5 строка 66-99 быстро пробежаться по примерам. Не все комментарии освежили память в рамках этого урока. Строка 108-119 не смог разобраться откуда появился данный метод у этого члена класса?

1. Урок 6 строка 69 опять class.
2. Урок 8 не получилось воспроизвести:

Copy-Item @HashArguments # будет то же что и

Copy-Item -Path "test.txt" -Destination "test2.txt" -WhatIf $true

Строка 23-24 добавляет один аргумент, а "testvalue" является его значением……)

$HashArguments.Add("testname","testvalue") # просто используем метод Add с **двумя** аргументами

1. Было бы неплохо привести более простой пример с использованием интерфейсов урок 9.

ООП в Павершелл, программа обучения по часам

часы тема

1 Общие сведения, знакомство с Powershell ISE

3 Типы данных в Powershell, операторы и их приоритеты

3 Циклы и ветвления

5 Функции, области видимости переменных

3 Массивы и строки

3 Объекты и классы

3 Регулярные выражения, шаблоны

3 Перегрузка операций и методов

2 Обработка исключений

2 Наследование и интерфейсы

2 Потоки и файлы

3 Многофайловые программы. Удаленное исполнение скриптов. Типы аутентификации.

3 Работа с rest-АРI

3 работа с АСL

3 иной полезный функционал .Net framework

итого