PowerShell

PowerShell Programming Control Flow

Preparation

Prepare a PowerShell script with extension .ps1 for each exercise in this series. Execute each script by navigating to the folder containing it. Depending on your system configuration use may need administrator privileges to do so.

Exercise 1

a. Create an array variable $proc containing the processes currently running on the system.

b. Print the 2nd, 5th and 10th entry from the array.

Exercise 2

a. Check whether the string Julius is identical to julius. At first use the standard comparison operator. Afterwards employ the case-sensitive comparison.

b. Prüfen Sie, ob *Julius* mit *julius* identisch ist. Verwenden Sie ein Mal die Unterscheidung von Groß- und Kleinschreibung und ein Mal nicht.

c. Check whether $\frac{14}{3}$ is greater or equal than 4.6 (4,6 in German decimal notation).

d. Figure out whether the value of the variable *$HOME* contains the letter ‘o‘. Find two approaches to solve the task.

Exercise 3

a. Store a number from the range 1 thru 10 in a variable. Print *number is greater than 5*, if the number is greater than five

b. Print *number is exactly 5*, if the number is five

c. Print *number is lesser than 5* in all other cases.

Exercise 4

Program a loop that prints the values 10 thru 100 with an increment of 10.

Use both a For- and a ForEach-loop to print the positive multiples of 5 up to 100 (inclusive).

PowerShell

PowerShell Programming Control Flow

Exercise 5

a. Schreiben Sie ein Skript, das einen kontinuierlichen Ping (wie unter GNU/Linux) durchführt. Die Wartezeit zwischen zwei Pings soll dabei 1 Sekunde betragen.

b. Entwickeln Sie Pseudo-Code, der beschreibt, wie eine Pipeline mit einer while-Schleife aus der bash in PowerShell umgesetzt werden kann. Achten Sie darauf, die Syntax möglichst wenig zu verändern.

Exercise 6

Write a script that determines the machine precision of your computer using a sequence of powers of 2. Return the smallest exponent that yields a result different from 0.

Exercise 7

Consider the following PowerShell dode:

```
gci | % { mi $_ .. }; $d=$(gl); sl ..; ri $d
```

a. Describe the functionality of the script as accurately as possible.

b. Write the code as a script using the appropriate long verb-noun-syntax of all Cmdlets and parameters used.

Exercise 8

Write your own custom version of the parameter -Include for Cmdlet Get-ChildItem. Iterate over the items in the current directory and only return those with a matching extension.

The extension can be provided hard coded in the script or in an advanced version may be passed as a command line argument.