

PowerShell - Windowsi skriptimiskeel

Andres Apevalov,
MT-IT MAG II

Miks on vajalik?

- Toob kokku opsüsteemi erinevad „jupid“ ühte keelde
- Admin ei pea tundma mitut keelt/tehnoloogiat
- Peidab erinevate tehnoloogiate keerukused kasutaja eest
- Annab sõbraliku süntaksi
- BAT PowerShell'ile on nagu koopamaalingud arenenud keelele

Realisatsioon

- Interpreteeritud keel
- Graafiline keskkond - peamiselt arenduseks
- Käsurida – töötamiseks
- Mõlemad tulevad ühe, allalaetava, pakina
- Hetkel versioon 2 *community tech preview* staadiumis
- Käivituspoliitika – keelatud, signeeritud, kõik lubatud

Keele elemente

- Muutujad: `$var = „tere“`
- Operaatorid: `-lt, -gt, -ne, !, -or, -bor, ...`
- Massiivid: `@(1, 2, 3)`
- Räsitabelid: `@{prop=val;prop1=val1}`
- Tingimused: `if-elseif-else, switch`
- Tsüklid: `while, for, foreach, do-until`

Vingemad asjad käsureal

- Funktsioonid: `function func {`
- Tüüpimine: `[int]$var = 0`; muidu „automaagiline“.
- Erindid: `throw-trap`
- Ja muid, vahvaid asju...
- CmdLet'id – `Read-Host`, `Write-Host`; saab ise toota ja paigaldada

Näited

#vähem-kui-geniaalne viis päevade eestindamiseks

- `$dayNames = @(@{est="E"; eng="Monday"}, @{est="T"; eng="Tuesday"},
@{est="K"; eng="Wednesday"}, @{est="N"; eng="Thursday"},
@{est="R"; eng="Friday"}, @{est="L"; eng="Saturday"},
@{est="P"; eng="Sunday"});`
- `Write-Host $weeks[$i].days[$k].value -foregroundcolor "red" -noNewLine;`
- `$tempDate = (Get-Date);` - kaks parsimise moodust: käsu- ja koodi (expression) jaoks.

Über-kalender

```
Windows PowerShell V2 (CTP)
E T K N R L P
13 1 2 3 4 5 6
14 7 8 9 10 11 12 13
15 14 15 16 17 18 19 20
16 21 22 23 24 25 26 27
17 28 29 30
PS C:\Documents and Settings\a\Desktop> _
```

Küsimused?