

Aufgabe 1

```
$n=50
For ($i = 1; $i -le 50; $i++) {
    Write-Progress -Activity "Suche..." -Status "$($i/$n*100)%"
    Start-Sleep -Milliseconds 100
}
```

Aufgabe 2

```
Function Show-WindirFiles {
    $files = gci $Env:WINDIR -File
    $files | ForEach-Object { Write-Progress -Activity "Finde Datei.." -Status "$_
gefunden"; Start-Sleep -Milliseconds 100 }
}
```

Aufgabe 3

```
a.
Function abs { param($n) If ($n -ge 0) { Return $n } Else { Return -1*$n } }
b.
Function sgn { param($n) If ($n -gt 0) { Return 1 } ElseIf ($n -eq 0) { Return 0
} Else { Return -1 } }
c.
Function fac { param($n) If ($n -eq 0) { Return 1 } Else { Return $n *
(fac($n-1)) } }
d.
Function bk { param($n,$k) Return fac($n)/(fac($k) * fac($n-$k)) }
e.
Function ack {
    param($n,$m)
    If ($n -eq 0) {
        Return $m+1 }
    ElseIf ($m -eq 0) {
        Return ack ($n-1) 1 }
    Else {
        Return ack ($n-1) (ack $n ($m-1)) }
}
```

Aufgabe 4

```
# only listing files, but not deleting them
Function Delete-FileTreeItems {
    param([string]$root)
    Get-ChildItem -Recurse -Force | ForEach-Object { If (-not $_.PSIsContainer) {
Write-Host "found $_" }
}
```

Aufgabe 5

```
# see file GroupMembership.ps1
```

Aufgabe 6

```
Function New-LocalLinuxUser {
    param([string]$UserName, [string]$Password)
    /usr/sbin/useradd $UserName --password $Password
}
```

Aufgabe 7

```
Function Upgrade-System {  
    param([switch]$Force=$False)  
    If (-not $Force) {  
        apt update && apt upgrade -y  
    }  
    Else {  
        apt update && apt list --upgradable | ForEach-Object { If (-not  
$_.StartsWith("Auflistung")) { apt install $($_.Split('/')[0]) -y } }  
    }  
}
```