Variablen, Datentypen, Formatierung

$i = 100	Datentyp automatisch **Int32**	$i Ausgabe: 100
$i = 100MB	Datentyp automatisch **Int64**	$i Ausgabe: 104857600
[int]$i=100	Datentyp **Int32**	explizite Typzuweisung: Fehlermeldung bei Falscheingabe bzw. wenn keine Konvertierung möglich
[int]$i="Test"	Fehler	Typ nicht konvertierbar
[int]$i=100MB	Fehler	Zahl zu groß
[int]$i=99.5	Datentyp **Int32**	$i.getType() > int32, Inhalt von $i: 100
$i="100"	Datentyp **String**	
Read-Host	Datentyp **String**	$s = read-Host "Eingabe"
[int]$i="100"	Datentyp **Int32**	Konvertierung möglich Zeichenketten können in Zahlen konvertiert werden, wenn sie ausschließlich Ziffern und Trennzeichen enthalten Anwendungsfall: [int]$i = read-Host "Zahl eingeben"

Array erzeugen

```
$array = 1,2,3    oder: $Orte="Hannover","Hamburg","Düsseldorf"
```

Array erweitern

```
$array += 4       oder: $Orte += „"
```

Array anzeigen

```
 $array          $Orte
 1               Hannover
 2               Hamburg
 3               Düsseldorf
 4               München
```

Einzelne Arrayelemente

```
$Orte[2]  >  Düsseldorf
```

Erstes Element `$array[0]`

Letztes Element `$array[-1]`

Zahlen formatieren

```
$a=67892423.456789
"{0:N2}" -f $a          >  67.892.423,46
"{0:0}" -f $a           >  67892423
"{0:0.000}" -f $a >  67892423,457
"{0:0,0.000}" -f $a     >  67.892.423,457
```

Zahlenformatierung bedeutet immer eine Umwandlung in einen String. Damit können auch weitere Zeichen hinzugefügt werden.

```
"1.Zahl: {0:N1}  2.Zahl: {1:0.00}" -f $a,1.234

Ergebnis:   1.Zahl: 67.892.423,5  2.Zahl: 1,23
```

Eindeutigen Pfad/ Dateinamen generieren:

```
"C:\Daten\LogFile-$env:COMPUTERNAME-{0:yyyyMMddHHmmss}.txt" -f (Get-Date)

>  C:\Daten\LogFile-DESKTOP-63P3KJL-20200426234555.txt
```

Datum

```
get-date -f "o"
```

ISO-Format: `2021-10-07T23:21:15.4386284+02:00`

Typ [math] >> Mathematische Funktionen

```
[math]::PI        >  3,14159265358979

$PI = [math]::PI

[math]::Round($PI,4)    >  3,1416
```

Berechnete Spalten erzeugen

In eine Zeile:

```
get-process a* |format-table
@{L="Name";E={$_.Name};width=30},@{Label="CPU(s)";Expression={"{0:N2}" -f
$_.cpu};align="right";width=10}
```

```
Name                                CPU(s)
----                                ------
Aac3572DramHal_x86                    0,31
Aac3572MbHal_x86                      0,02
Aac3572MbHal_x86                    489,20
AacKingstonDramHal_x64                0,03
AacKingstonDramHal_x86                0,14
AcPowerNotification                   0,42
ApplicationFrameHost                  0,16
ArmouryCrate.Service                  2,63
ArmouryCrate.UserSessionHelper        3,03
ArmourySocketServer                   0,31
ArmourySwAgent                        0,20
ArmouryWebBrowserEdge                 0,03
asus_framework                        0,47
asus_framework                        0,27
asus_framework                        0,22
asus_framework                        5,05
AsusCertService                      83,91
AsusFanControlService                 4,66
AsusUpdateCheck                       0,02
atkexComSvc                           5,45
audiodg                               2,05
```

```
get-process w* |select
Name,@{Label="CPU(s)";Expression={[math]::round($_.cpu,2)}}
```