

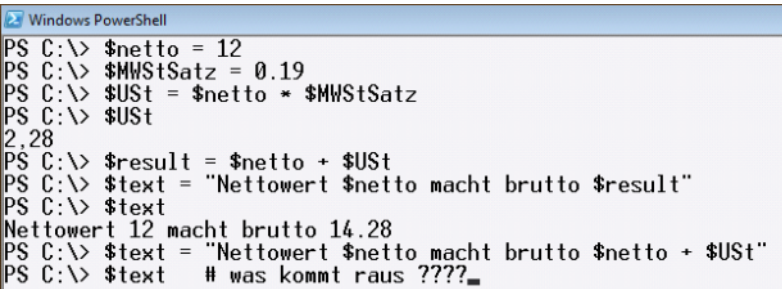
# Variable als Konzept

- Notwendigkeit von Variablen

Z

- Notwendigkeit von Datentypen

E



```
Windows PowerShell
PS C:\> $netto = 12
PS C:\> $MWStSatz = 0.19
PS C:\> $USt = $netto * $MWStSatz
PS C:\> $USt
2.28
PS C:\> $result = $netto + $USt
PS C:\> $text = "Nettowert $netto macht brutto $result"
PS C:\> $text
Nettowert 12 macht brutto 14.28
PS C:\> $text = "Nettowert $netto macht brutto $netto + $USt"
PS C:\> $text
# was kommt raus ???
```

- Beispielhafte Datentypen

i

- Ergebnis von Datentypen und darauf wirkende Operatoren

\

- Casten

- Konstante

V

- Automatische Powershell-Variablen

- Umgebungsvariablen

D

- Gültigkeit von Variablen

## Zusammenfassung

- Variablen speichern jede Art von Information
- Sie beginnen immer mit dem "\$"-Zeichen. Anschließend können Sie aus alphanumerischen und Sonderzeichen (z.B. Underscore) bestehen. Reservierte Worte dürfen nicht benutzt werden

- Groß- und Kleinschreibung spielt keine Rolle
- "Konstante" Variablen ändern ihren Wert nicht mehr
- **Powershell** hat schon selbst gewisse Variablen definiert (automatische Variablen)
- **Powershell** sorgt dafür, dass der Typ einer Variable zum Typ des zu speichernden Wertes passt (weakly typing)
- **Powershell** unterstützt Strong-Typing durch Voranstellen des Datentyps in eckigen Klammern []
- **Powershell** erstellt die Variablen jeweils innerhalb eines gewissen Gültigkeitsbereichs (local, private, script, global)
- An Umgebungsvariablen kann durch \$env: zugegriffen werden