LUZERN

KANTON LUZERN

06_PS_Funktionen_II

Modul 122

Abläufe mit einer Scriptsprache automatisieren

Berufsbildungszentrum
Wirtschaft, Informatik und Technik
bbzw.lu.ch

Mitteilungen

22.04.2023

- Heute keine Mitteilungen

Handlungsnotwendige Kenntnisse

HANOK	Kenntnisse
1.6	Kennt das Vorgehen zur Realisierung von Scripts in der Systemadministration.
2.1	Kennt grundlegende Funktionalitäten der eingesetzten Scriptsprache.
4.1	Kennt ein Testverfahren für Scripts.
5.1	Kennt die Elemente einer Dokumentation für die involvierten Rollen (z.B. System, Administrator, Entwickler)

Themen

- Funktionen: Rückgabewerte
  (Herdt-Buch: Kap 8.2 - 8.5, S. 114 – 117)

Auftrag 1 (10‘)

1) Vollziehen Sie die Berechnung der Mehrwertsteuer nach, indem Sie die nachfolgende Funktion übernehmen und mit verschiedenen Werten ausprobieren.

```
function MwSt($Betrag, $Satz)
{
  $Betrag / 100 * $Satz
}
```

Prüfen Sie folgende Eingaben und überlegen Sie was ausgeführt wird

a) `MwSt 1000 7.7`
b) `MwSt -Betrag 1000 -Satz 7.7`
c) `MwSt`

2) Nun würden wir gerne den Satz von 7.7% als Default Wert speichern, dass der User dies nicht immer angeben muss, wenn er was mit 7.7% rechnen muss. Dazu setzen wir einen Default-Wert, dieser kann jeder Zeit überschrieben werden.

```
function MwSt($Betrag, $Satz = 7.7)
```

Auftrag 1 (10‘)

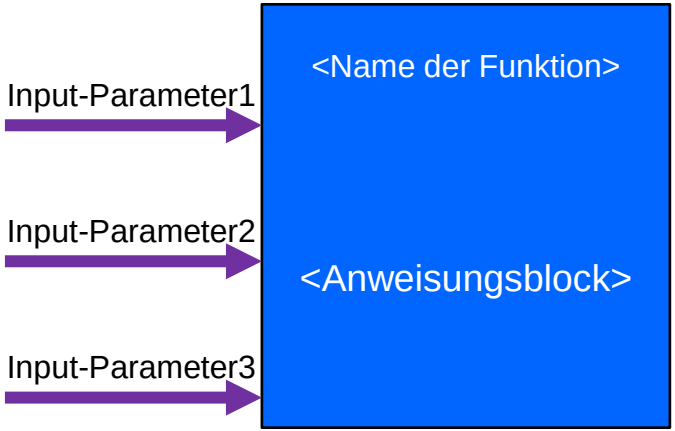
3) Um sicherzustellen, dass der User immer Zahlen eingibt, werden wir die Datentypen noch definieren. Dazu verwenden wir den Datentyp `[double]`.

```
function MwSt([double]$Betrag, [double]$Satz)
{
  $Betrag / 100 * $Satz
}
```

4) Prüfen Sie Get-Help MwSt die erstellte Funktion

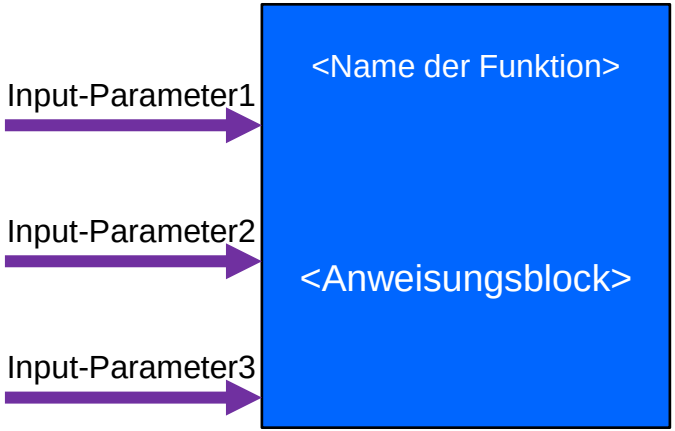
Funktionen Syntax – OptionA

```
Function <Name der Funktion> ($Input1, $Input2, $Input3)
{
   <Anweisungsblock>
}
```



Funktionen Syntax – OptionB

```
Function <Name der Funktion>
{
 param (
         $Input1,
         $Input2,
         $Input3
         )
<Anweisungsblock>
}
```



Darstellung von Parametern in der Definition der Funktion

Folgende Darstellung haben in der PS die gleiche Bedeutung.
Empfehlung: Verwenden sie diejenige die Ihnen mehr zusagt.

```
function fu (
    $a, #Kommentar zu a
    $b  #Kommentar zu b
    )
{
    write-host "params are: " $a $b
}
```

Option A

```
function fu2 {
  param (
    $a, #Kommentar zu a
    $b  #Kommentar zu b
    )

    write-host "params are: " $a $b
}
```

Option B

Aufgabe A (mPreis)

Erarbeiten Sie in den unten folgenden Punkten schrittweise eine Funktion («M122_06_AufgabeA _<Name>_<Vorname>.ps1»), die den Preis mehrerer Stücke eines Artikels berechnet. Diese Funktion soll insgesamt drei Parameter kennen:
Den Stück-Preis, die Anzahl Stück und den Artikelname.

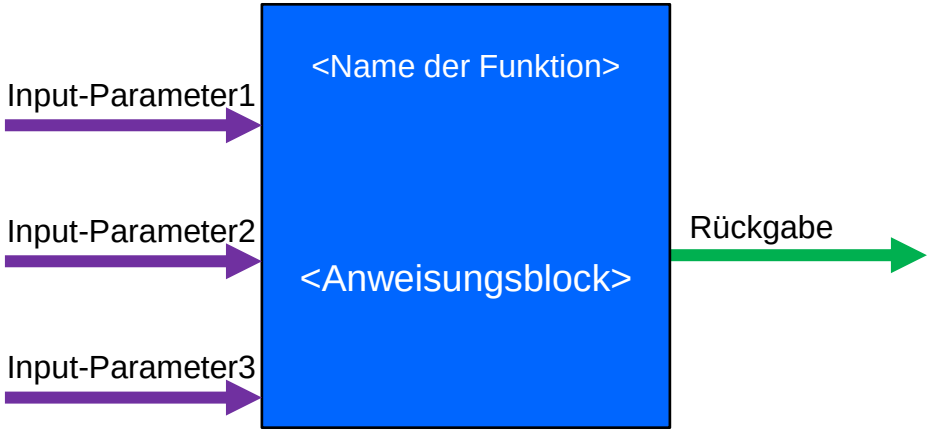
1) Schreiben Sie zuerst eine Funktion mPreis2, die den Stück-Preis und die Stückzahl akzeptiert.
   Geben Sie das Resultat mit `echo` aus, z.B.:
   ```
   echo "Der Preis beträgt: $Preis"
   ```
2) Schreiben Sie dann die Funktion mPreis3 mit 3 Parametern und geben Sie das Resultat aus, z. B.:
   ```
   echo "Der Preis von $Artikel beträgt: $Preis"
   ```

Aufgabe A (mPreis)

4) Schreiben Sie dann eine Funktion mPreis4 mit denselben Parametern und Vorgabe der Typen des Parameters, Bsp. mPreis4 ([int] $anz, ….)
5) Geben Sie dann versuchshalber bei den entsprechenden Parametern einmal eine Zeichenkette als Zahl [int] und einmal eine Zahl als Zeichenkette [string] ein. Weshalb resultiert nur einmal ein Fehler?
6) Ergänzen sie danach mPreis4 mit einem Kommentar zu jedem Parameter. Schreiben sie die Parameter zuerst jeden einzelnen auf eine separate Zeile und hängen sie hinten jeweils einen Kommentar an.

Funktionen Syntax

```
Function <Name der Funktion> ($Input1, $Input2, $Input3)
{
   <Anweisungsblock>
   return $Rueckgabe
}
```



Aufgabe B (Funktion mit Rückgabe)

- Verwenden Sie eine neue PS-Skriptdatei («M122_06_AufgabeB_<Name>_<Vorname>.ps1»).
- Kopieren sie die Beispiel-Funktion „Get-MReturn“ (siehe unten) in Ihre PS-Datei.
- Führen Sie die Funktions-Definition in der ISE aus
- Rufen Sie in der Konsole `Get-MReturn` auf, und weisen sie das Resultat der Variablen `$vv` `$vv = Get-MReturn`
- Was ist das Resultat? Wie können Sie es überprüfen?

```
function Get-MReturn
{
   $ww =100
   return $ww
}
```

Aufgabe C (mSumRet: Funktion mit Parameter & Rückgabe)

- Verwenden Sie eine Datei „M122_06_ AufgabeC _<Name>_<Vorname>.ps1“. Arbeiten sie mit der ISE.
- Schreiben Sie eine Funktion mSumRet, die 2 ganze Zahlen als Inputparameter akzeptiert, und die deren Summe als Ergebnis (=Rückgabe) zurückgibt
- Verwenden Sie in der Funktion für die Parameter die Schreibweise mit `param ( … hier die Parameter …),` jeder Parameter auf eigener Zeile und ergänzen Sie für jeden Parameter einen passenden Kommentar.
- Ergänzen Sie unter den Parametern einen Kommentar für die Rückgabe

Aufgabe C (mSumRet: Funktion mit Parameter & Rückgabe)

- Test1: „Testen“ Sie ihr `mSumRet` folgendermassen:
  - `mSumRet 3 4` # dies muss als Ergebnis 7 ergeben
  - Nehmen Sie die Tests in Ihre Datei auf. Und erläutern Sie den Test (was wurde gemacht, was ist Ergebnis) in einem Kommentar
- Test2: „Testen“ sie `mSumRet` mit:
  - `mSumRet 3 (mSumRet 100 100)` # dies muss als Ergebnis 203 ergeben

Aufgabe C (mSumRet: Funktion mit Parameter & Rückgabe)

Zusatzaufgabe

- Schreiben Sie eine neue Funktion jeweils für
  - Eine Subtraktion
  - Eine Multiplikation
  - Eine Division

Hausaufgaben

- Fragen Sie Ihre Lehrperson

Vielen Dank für Ihre Aufmerksamkeit!

KANTON LUZERN

Berufsbildungszentrum Wirtschaft, Informatik und Technik BBZW

www.bbzw.lu.ch