

Bash-Programmierung

Angabe der Shell

```
#!/bin/bash
```

Variablen

Wertzuweisung

```
VARIABLE=2 (nur ganze Zahlen)
VARIABLE=a
VARIABLE=Hallo
VARIABLE=Hallo\ Welt!
VARIABLE="Hallo Welt!"
VARIABLE=$(pwd)
VARIABLE=`pwd`
```

Einlesen von Tastatur

```
read VARIABLE
```

Vordefinierte bash-Variablen

```
$? Rückgabewert des letzten Kommandos (0=erfolgreich, 1=Fehler)
 $# Anzahl der dem Programm übergebenen Argumente
 $0 Dateiname des gerade ausgeführten Shell-Skriptes
 $1 .. $9 Parameter 1 bis 9
 $* Alle übergebenen Parameter
 $PATH Pfadliste
 $HOSTNAME Name des Rechners
 $HOME Pfad des home-Verzeichnisses
 $LOGNAME username
```

Bedingungen

0 = true, 1 = false

Zeichenketten

```
[ $VARTEXT ] VARTEXT nicht leer
[ -z $VARTEXT ] VARTEXT leer
[ $VARTEXT1 = $VARTEXT2 ] VARTEXT1 und VARTEXT2 identisch
[ $VARTEXT1 != $VARTEXT2 ] VARTEXT1 und VARTEXT2 nicht identisch
```

Zahlen

```
[ $VARZAHL1 -eq $VARZAHL2 ] (==)
[ $VARZAHL1 -ne $VARZAHL2 ] (!=)
[ $VARZAHL1 -gt $VARZAHL2 ] (>)
[ $VARZAHL1 -ge $VARZAHL2 ] (>=)
[ $VARZAHL1 -lt $VARZAHL2 ] (<)
[ $VARZAHL1 -le $VARZAHL2 ] (<=)
```

Dateien

```
[ -e $DATEI ] Datei existiert
[ -f $DATEI ] ist ,normale' Datei
[ -r $DATEI ] Datei lesbar
[ -w $DATEI ] Datei schreibbar
[ -x $DATEI ] Datei ausführbar
[ $DATEI1 -nt $DATEI2 ]
Datei1 ist neuer als Datei2
```

Verknüpfung von Bedingungen

```
[ Bedingung1 ] || [ Bedingung2 ] ODER-Verknüpfung
[ Bedingung1 ] && [ Bedingung2 ] UND-Verknüpfung
! [ Bedingung ] NICHT-Verknüpfung
```

Verzweigungen

if

```
if Bedingung
then
    Kommandos
else
    Kommandos
fi
```

Beispiel

```
echo -n "Zahl1: ";read Z1
echo -n "Zahl2: ";read Z2
if [ $Z1 -eq $Z2 ]
then
    echo $Z1 == $Z2
else
    echo $Z1 != $Z2
fi
```

case

```
case Wort in
    Muster1) Kommandos ;;
    Muster2) Kommandos ;;
    Muster3|Muster4) Kommandos ;;
    * ) Kommandos ;;
```

esac

Beispiel

```
echo -n "Name: ";read NAME
case $NAME in
    Anton) echo Servus! ;;
    Berta|Carla) echo Küß die Hand! ;;
    *) echo Wer bist Du? ;;
```

esac

Schleifen

for

```
for Name in Wort1 Wort2 ...
do
    Kommandos
done
```

Beispiel 1

```
for i in 1 2 3 4 5
do
    echo $i
done
```

Beispiel 2

```
n=1
for i in $(ls)
do
    echo $n.Datei: $i
    n=$((n+1))
done
```

while

```
while Bedingung
do
    Kommandos
done
```

Beispiel

```
i=1
while [ $i -le 5 ]
do
    echo $i
    i=$((i+1))
done
```

until

```
until Bedingung
do
    Kommandos
done
```

Beispiel

```
i=1
until [ $i -gt 5 ]
do
    echo $i
    i=$((i+1))
done
```

Funktionen

Definition

```
funktionsname ()
{
    Kommandos
}
```

Beispiel

```
summe ()
{
    echo ${1+$2}
}
```

Aufruf

```
funktionsname Argumente
```

Beispiel

```
summe 2 3
```

Skript beenden

```
exit 0
```