Shellprogrammierung Snippets

Jan Caspar, Aktualisiert 1. Juli 2017, v 1.0.0

Built-In Variablen

$#	Anzahl der Argumente die übergeben wurden.
$@	Alle Vraiblen die Übergeben wurden (mit Leerzeichen getrennt)
$?	War der letzte Befehl erfolgreich, 0 = Ja
$$	Prozess ID des Scripts
$1,$2,. . . $n	die einzelnen Argumente

Abschneiden von Mustern

${variable%muster} Entfernt rechts das kleinste passende Stück.

${variable%%muster} Entfernt rechts das größte passende Stück

${variable#muster} Entfernt links das kleinste passende Stück.

${variable##muster} Entfernt links das größte passende Stück

Beispiel

```
pfadname="/var/www/index.html"
echo $pfadname   #/var/www/index.html
echo "Pfad: ${pfadname%/*}"   #Pfad: /var/www
echo "Dateiname: ${pfadname##*/}" #Dateiname: index.html
```

Umgebungsvariablen

HOME	Enthält das Homeverzeichnis des Benutzers.
PATH	Suchpfad für Kommandos.
PWD	Enthält das Aktuelle Verzeichnis.
?	Enthält den Exitstatus des letzten Kommandos

Standardeingabe

```
#!/bin/bash
#Einfaches einlesen
read -p "Geben sie ihren Namen ein:" name
echo Hallo: $name

#Optionen abfragen
read -p "A (a) oder B (b)?, Abbruch anderen Taste" kommando;
if [ $kommando == 'a' ];
        then
                starte_programm_a;
        elif [ $kommando == 'b' ];
                then starte_programm_b;
        else
                echo "Abbruch.";
fi
```

Files checken

```
#!/bin/bash

if [ -f /home/user/test.txt ]
then
        echo "File exists."
elif [ -d /home/user/test.txt ]
then
        echo "File is a directory."
        exit 1
else
        echo "File not found."
        exit 1
fi
```

Parameteranzahl <> 0

```
#!/bin/bash

if [ $# -eq 0 ]
then
        echo "Args missing!"
exit 1
```

IFS wechseln

```
#!/bin/bash
IFSAVE=IFS
IFS=';'
#do stuff
IFS=IFSAVE
```

Textdatei zeilenweise verarbeiten

```
#!/bin/bash
#$1 should contain file
#might need to adapt IFS
while read x y z
do
        echo "processing $x $y $z"
        res="${x} @ testscript"
done < $1
```

Beispiel: Wetter

Wetterevent aus

```
04.05.2017 - 16:22 Uhr
Hagel in Brunnhof (Amstetten)
04.05.2017 - 16:20 Uhr
Hagel in Miesenbach bei Birkfeld (Weiz)
```

suchen

```
#!/bin/bash
if [ ! -f ./wetter ]
then
 echo "File wetter not found"
 exit 1
fi
if [ $# -ne 1 ]
then
 echo "Wrong number of arguments!"
 exit 2
fi
R=`grep -Ec "^$1 " wetter`
if [ $R -eq 0 ]
then
 echo "Wetterereignis $1 ist nicht vorgekommen!"
 exit 3
fi
rm ./output 2> /dev/null
rm ./tmp 2> /dev/null
grep -E "^$1 " wetter > ./tmp
echo "$1 gab es in folgenden Orten:" > ./output
while read v1 v2 v3
do
 O=`echo $v3 | cut -f1 -d''`
 B=`echo $v3 | cut -f2 -d''`
 B1=`echo $B | sed -r 's/\)//g'`
 echo $O im Bezirk $B1 >> output
done < ./tmp
cat ./output
```

Beispiel: Skigebiete

Schreiben Sie ein Shell Script formatieren, dass den File schi_gebiet entsprechend der Darstellung unten umformatiert und im File schigebiet_2 speichert.

```
Großglockner Resort;13;€ 45 (Tageskarte Hauptsaison)
Matrei in Osttirol, Kals am Großglockner, Großdorf, Lesach, Virgen
Lermoos – Grubigstein;8;€ 43 (Tageskarte Hauptsaison)
```

1. Der File schi_gebiet wurde unter Windows generiert.
2. Überprüfen Sie, ob der File schi_gebiet wirklich existiert.
3. Pro Schigebiet sind zwei Zeilen vorzusehen. In der ersten Zeile sind Name des Schigebiets, Anzahl der Lifte und Preis der Tageskarte vorzusehen. Die einzelnen Felder sind durch Strichpunkte zu trennen. In der zweiten Zeile sind die am Schigebiet beteiligten Ortschaften zu listen, die jeweils durch Beistriche voneinander zu trennen sind.
4. Die Ortschaften pro Schigebiet sind zusätzlich einzeln in den File gemeinde zu schreiben. Dieser File soll die Ortschaften aller Schigebiete beinhalten, er ist vor Ende des Shell Scripts absteigend sortiertauszugeben.

```
#!/bin/bash
if [ ! -f ./schi_gebiet ]
then
 echo "File ./schi_gebiet missing..."
 exit 1
fi
rm ./schigebiet_2 ./gemeinde 2> /dev/null
while read line
do
 read ort
 out2=`echo $ort | cut -f2 -d";"`
 read lift
 out3=`echo $lift| cut -f2 -d";" `
 read karte
 out6=`echo $karte | cut -f2 -d";"`
 echo "${line};${out3};$out6" >> schigebiet_2
 echo "$out2" >> schigebiet_2
 read leerzeile
 IFSAVE=$IFS
 IFS=","
 for ort in $out2
 do
 echo $ort >> gemeinde
 done
 IFS=$IFSAVE
done < ./schi_gebiet
echo
echo
cat schigebiet_2
echo
echo
sort gemeinde
```

Beispiel: mmv

Erstellen Sie ein Shellscript mmv (=multiple move), das die Extensions von Filenamen, die sich in im aktuellen Arbeitsverzeichnis befinden, ändert. Überprüfen Sie auch die Anzahl der Argumente.

```
#!/bin/bash
#args prüfen!
for f in *.$1
do
 if [ -f $f ]
 then
        mv ./"$f" "${f%$1}$2"
 fi
done
```