- Bitte tragen Sie am unteren Blattrand Ihren Namen, Lehrgang und das heutige Datum ein.
- Dieses Übungsblatt ist zum Ende des jeweiligen Praktikums bei der Praktikumsbetreuung als Nachweis abzugeben.
  Sie können sich eine Kopie dieses Blattes unter ftp://lunabox.de/fh-isny/13.Info/ oder unter **http://lunabox.de/13.info** herunterladen.

1. **Variablen** können in den gängigen UNIX-/Linux-Shells (Linux-Standard: Bash) ohne explizite Deklaration initialisiert werden. Beispiele: **`var1=irgendwas ; klein=sowas ; GROSS="was anderes" ; z1=13 ; z2=234`**
   Der Zugriff auf den Wert einer Variablen erfolgt dann durch das Voranstellen eines $-Zeichens.
   Beispiel: **`echo "$var1 - $klein - $GROSS - $z1 - $z2"`**

2. Bestimmte Variablen – **Umgebungsvariablen** – werden vom System (der Shell) gesetzt. Mit dem Befehl **`env`** kann man sich diese anzeigen lassen – unter anderem wird man hier auch den eigenen Benutzernamen (USER) und sein Homeverzeichniss (HOME) finden. Sehen Sie sich Ihre aktuellen Umgebungsvariablen genauer an.

3. Man kann Shell-Variablen nicht nur direkt Werte zuweisen, sondern auch durch die sog. **Kommandosubstitution** die Ausgabe eines Befehlsaufrufes als Wert einer Variablen speichern.
   Folgende Syntax ist mit den meisten UNIX-Shells kompatibel: **`variable=´kommando´`** (Vorsicht: ´-Zeichen)
   In der Bash - im Gegensatz zur Korn-Shell/ksh (AIX) - ist auch **`variable=$(kommando)`** möglich.
   Beispiele: **`homelisting=$(ls -lah ~/) ; echo -e "Home-Listing von $USER: \n $homelisting"`**
   `mounts=$(mount) ; echo -e "MOUNTS:\n-------\n$mounts"`

4. Jede ASCII-Datei kann mit **`chmod`** ausführbar gemacht werden, und wird somit bei jedem direkten Aufruf (im aktuellen Arbeitsverzeichniss mit vorangestelltem **`./`**) automatisch von einer Shell – im Regelfall unter Linux der Bash – interpretiert. Um die Wahl des Interpreters (z.B. die ksh oder auch perl, ruby, ...) festzuschreiben wird das sog. Shebang verwendet – die Datei beginnt dabei mit dem folgenden String in der ersten Zeile:
   `#!/bin/bash` oder z.B. **`#!/usr/bin/perl`**

5. Innerhalb von Shells(cripts) hat man Zugriff auf folgende Variablen:

`$0`	Name des Shellsciptes (mit vollem Pfad)
`$1 / $2`	1. / 2. Positionsparameter
`$* / $@`	Alle Positionsparameter (als ein String: "$1 $2 $3..." / als einzelne Strings: "$1" "$2" "$3" "...")
`$#`	Anzahl der Positionsparameter
`$$`	Prozessnummer der aktuellen Shell
`$?`	Rückgabewert des direkt zuvor ausgeführten Kommandos (0=erfolgreich)

   Testen Sie kurz zum besseren Verständnis folgendes Script (z.B. als Datei "**~/s1**") mit untenstehendem Aufruf:

```
#!/bin/bash
echo "Scriptname + Parameter($# Stueck): $0  $*"
echo "PID der aktuellen Shell: $$"
echo "Parameter1: $1 ; Parameter2: $2 "
```

   Aufruf: **`cd ~ ; ./s1 par1 "par2 - und so weiter ..."`**

6. Der **`test`**-Befehl kann genutzt werden, um verschiedenste Tests auszuführen. Lesen Sie hierzu die man-page. Beachten Sie die zwei möglichen Aufrufmöglichkeiten:
   `test` `-optionen parameter`
   `[` `-optionen parameter` **`]`** (eckige Klammern anstatt **`test`** ; wichtig: Leerzeichen um die Klammern!)
   Beispiel: **`echo -n "Zahl: " ; read a ; b=5 ; [ $a -gt 5 ] ; echo "$? (wenn 0: groesser $b)"`**

   Mit **`Befehl1 && Befehl2`** wird der Befehl2 nur ausgeführt, wenn Befehl1 **erfolgreich** war (`$? = 0`),
   mit **`Befehl1 || Befehl2`** wird der Befehl2 nur ausgeführt, wenn Befehl1 **nicht erfolgreich** war (`$? != 0`)
   So lässt sich z.B. der test-Befehl recht effizient einsetzen: **`dir="~/xyz" ; [ -d $dir ] || mkdir $dir`**

7. In der Bash gibt es auch die aus anderen Programmiersprachen bekannten Kommandos zur Ablaufsteuerung, so dass auch komplexere Shellscripte möglich sind:

```
if Ausdruck
 then Kommando1
 else Kommando2
fi
```

```
case var in
  muster1|m1) Kommando1 ;;
  muster2|m2) Kommando2 ;;
  default)    KommandoX ;;
esac
```

```
while Ausdruck
do Kommando1
   Kommando2
done
```

```
until Ausdruck
do Kommando1
   Kommando2
done
```

```
for var in 1 2 3 n
do Kommando1
   Kommando2
done
```

   Genaueres dazu (und auch zum hier vernachlässigten select-Kommando) finden Sie in der man-page zur Bash ab ca. Zeile 250 – lesen Sie diesen Abschnitt aufmerksam und testen einige einfache eigene Konstrukte (zumindest **`if`**, **`case`** und **`for`**) selbstständig mit sinnvollen Beispielen. Verwenden Sie einen **`test`** als `Ausdruck`.

8. Löschen Sie bitte alle in diesem Praktikum erstellten Dateien und Verzeichnisse. Danke.

Nachname, Vorname	Lehrgang	Datum	Unterschrift StudentIn	Unterschrift Betreuer