Snippets

- Zufallszahlen
- Prüfen ob eine Variable einen Integerwert enthält
- mit Unicode-Zeichen malen
- Datum/Zeit-Berechnung
- Anzahl verschiedener Dateitypen zählen

Zufallszahlen

Zufallszahlen aus /dev/urandom mit einem bestimmten Bereich erzeugen

```
# Zahlen von 1 - 10
RND=`od -vAn -N1 -tu1 < /dev/urandom` && echo $(( $RND % 10 + 1 ))
```

Prüfen ob eine Variable einen Integerwert enthält

```
#!/bin/bash

var="test"
#var=5

if [[ $var =~ ^-?[0-9]+$ ]]; then
  echo "$var ist int"
else
  echo "$var ist kein int"
fi
```

hat diese Ausgabe:

```
$ bash -x test
+ var=5
+ [[ 5 =~ ^-?[0-9]+$ ]]
+ echo '5 ist int'
5 ist int

$ bash -x test
+ var=test
+ [[ test =~ ^-?[0-9]+$ ]]
+ echo 'test ist kein int'
test ist kein int
```

mit Unicode-Zeichen malen

Die Codetabelle gibts hier: http://unicode-table.com/de/

Verwenden läßt sich das dann wie folgt:

```
$ echo -e '\u00A9'
©
$ printf '\u00A9\n'
©
```

Damit lassen sich dann z.B. auch Spielfelder, Dialogboxen und ähnliches zeichnen (wobei für Dialoge das Tool „dialog“ besser geeignet wäre).

Datum/Zeit-Berechnung

Datum und Zeit-Berechnung und -Manipulation mit BASH-Boardmitteln ist leider sehr mühselig. Das GNU date-Kommando kann uns hier sehr helfen:

Bei allen Beispielen gilt:

`DATUM="2009-02-25 10:30:38"`

Ein Datum in UNIX-Timestamp umwandeln:

`STAMP=`date --utc --date "$DATUM" +%s`

Und wieder zurück:

`date --utc --date "1970-01-01 $STAMP sec" "+%Y-%m-%d %T"`

Sekunden/Minuten/Stunden/Tage zwischen zwei Daten ausgeben:

```
dateDiff (){
    case $1 in
        -s)   sec=1;      shift;;
        -m)   sec=60;     shift;;
        -h)   sec=3600;   shift;;
        -d)   sec=86400;  shift;;
        *)    sec=86400;;
    esac
    dte1=$(date2stamp $1)
    dte2=$(date2stamp $2)
    diffSec=$((dte2-dte1))
    if ((diffSec < 0)); then abs=-1; else abs=1; fi
    echo $((diffSec/sec*abs))
}

# Beispiel:
# -s in sec. | -m in min. | -h in hours  | -d in days (default)
 dateDiff -s "2006-10-01" "2006-10-32"
 dateDiff -m "2006-10-01" "2006-10-32"
 dateDiff -h "2006-10-01" "2006-10-32"
 dateDiff -d "2006-10-01" "2006-10-32"
 dateDiff  "2006-10-01" "2006-10-32"

# number of seconds between two times
```

```
 dateDiff -s "17:55" "23:15:07"
 dateDiff -m "17:55" "23:15:07"
 dateDiff -h "17:55" "23:15:07"

# number of minutes from now until the end of the year
 dateDiff -m "now" "2006-12-31 24:00:00 CEST"
```

weitere Standardfeatures von date, die nicht sehr ausführtlich dokumentiert sind:

```
# add 2 days, one hour and 5 sec to any date
date --date "$DATUM  2 days 1 hour 5 sec"

# substract from any date
date --date "$DATUM 3 days 5 hours 10 sec ago"
date --date "$DATUM -3 days -5 hours -10 sec"

# or any mix of +/-. What will be the date in 3 months less 5 days
date --date "now +3 months -5 days"

# time conversions into ISO-8601 format (RFC-3339 internet recommended format)
date --date "sun oct 1 5:45:02PM" +%FT%T%z
date --iso-8601=seconds --date "sun oct 1 5:45:02PM"
date --iso-8601=minutes

# time conversions into RFC-822 format
date --rfc-822 --date "sun oct 1 5:45:02PM"
```

Anzahl verschiedener Dateitypen zählen

manchmal sehr praktisch:

```
/tmp$ for i in `find -name "*.*" | sed 's/.*\.\(.*\)$/\1/' | grep . | sort -uf`; do echo "$i: `find -name \"*.$i\" | wc -l`"; done
pdf: 11
pub: 1
txt: 3
```