

video/audio-delay in Videos mit ffmpeg korrigieren

Fall 1: audio beginnt vor video (150 ms = 0.15 s)

```
ffmpeg -i video.mp4 -itsoffset 0.150 -i video.mp4 -vcodec copy -acodec copy -map 0:0 -map 1:1 video_insync.mp4
```

Fall 2: video beginnt vor audio (150 ms = 0.15 s)

```
ffmpeg -i video.mp4 -itsoffset 0.150 -i video.mp4 -vcodec copy -acodec copy -map 0:1 -map 1:0 video_insync.mp4
```

WiFi-QRCode mit qrencode

```
$ qrencode -t SVG -o wifi.svg "WIFI:S:mySSID;T:WPA2;P:myPass;"
```

Individuelle body id in typoscript

```
# überschreibt den gesetzten BODY-Tag
page.bodyTag >
# CObject vom Typ TEXT
page.bodyTagCObject = TEXT
# Die ID mit der die Seite gespeichert wird wird genommen (hierum wrappen wir in der nächsten Zeile)
page.bodyTagCObject.field = uid
page.bodyTagCObject.wrap = <body id="meineid-|">
```

send-hook mit Datum in mutt

```
set record=+sent-`date +%Y`
```

ergibt sent-2013

```
set record=+sent-`date +%Y-%m`
```

ergibt sent-2013-01

countdown-script

countdown script

[countdown](#)

```
#!/bin/bash
#
#####
#####
#
# countdown : bash script to visually count down a specified number of
seconds
#
#####
#####
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of the GNU General Public License
# as published by the Free Software Foundation; either version 2
# of the License, or (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA
02111-1307, USA.
#
#####
#####
#
MY_NAME=countdown
MY_VERSION=1.1
MY_RELDATE=2000-07-06
MY_AUTHOR_NAME="Roy Batty"
MY_AUTHOR_MAIL="roybatty@altavista.net"
#
#####
#####
#
STAMP=/tmp/.stamp.$USER.$$
#
function init_timer() {
    [ -f $STAMP.stop ] && rm -f $STAMP.stop 2>/dev/null
    echo S=`date +%s` >$STAMP
```

```

}
function run_timer() {
    while [ ! -f $STAMP.stop ]; do
        echo S=`date +%s` >$STAMP
        sleep 1
    done
    rm -f $STAMP $STAMP.stop
}
function stop_timer() {
    touch $STAMP.stop
}
function clean_exit() {
    stop_timer
    echo " "
    echo "Terminated."
    mpv $HOME/.bin/Alarmclock-mechanical.ogg
    trap "" 0 1 3 5 9 15
    exit 0
}
function syntax_error() {
    cat <<-EOF 1>&2

    $MY_NAME $MY_VERSION ($MY_RELDATE) - bash script to visually count
down
    Copyright 2000: $MY_AUTHOR_NAME <$MY_AUTHOR_MAIL>
    Usage:

    $0 [[h] m] s
        counts down h hours, m minutes and s seconds

    EOF
    exit 1
}
function get_diff() {
    let DIFF=0
    [ "$1" = "" -o "$4" != "" ] && syntax_error
    while [ "$1" != "" ]; do
        let p=0
        [ "$1" != "0" ] && ! let p=$1 2>/dev/null && syntax_error
        [ $p -lt 0 ] && syntax_error
        let DIFF=$DIFF*60+$p
        shift
    done
}
function display_nice() {
    local s=$1; local h=0; local m=0; ni=""
    [ $s -ge 3600 ] && let h=$[ s/3600 ] && let s=$[ s%3600 ]
    [ $s -ge 60 ] && let m=$[ s/60 ] && let s=$[ s%60 ]
    [ $h -lt 10 ] && ni="0"; ni="$ni$h:"
    [ $m -lt 10 ] && ni="{ni}0"; ni="$ni$m:"
    [ $s -lt 10 ] && ni="{ni}0"; ni="$ni$s"
}

```

```
    echo -n "$ni"
}
#
# main script
#
get_diff $*
trap clean_exit 0 1 3 5 9 15
echo "$MY_NAME $MY_VERSION ($MY_RELDATE) - Counting down `display_nice
$DIFF` [hrs:min:sec]"
if [ $DIFF -le 0 ]; then
    echo "Nothing to do." 1>&2
    exit 2
fi
echo " Elapsed | Remaining. | Complete"
let START=`date +%s`
declare -a HASH=( [0]=". " [1]="o " [2]="0 " [3]="- " )
compl="#####"
blank=" "
let ha=0
init_timer
run_timer &
while true; do
    . $STAMP
    let R=$DIFF-$S+$START
    if [ $R -gt 0 ]; then
        echo -n "${HASH[$ha]}`display_nice $[ $DIFF-$R ]` | `display_nice
$R` | "
        let P=$(( ($S-$START) * 100 / $DIFF )
        echo -ne "[${compl:0:P/2}${blank:0:50-P/2}] $P%\r"
    else
        echo -n "${HASH[$ha]}`display_nice $DIFF` | `display_nice 0` | "
        echo -ne "[${compl}] 100%\r"
        clean_exit
    fi
    let ha=$(( ha+1 ) % 4 ]
    sleep 1
done
```

Checksummen prüfen

```
md5sum -c md5sums 2> /dev/null | grep OK
```

Hash von Passwort bilden und nach stdout geben

```
$ echo -n sehrgeheim | sha256sum
```

From:

<https://g6r.de/dw/> - **g6r**

Permanent link:

<https://g6r.de/dw/tmp?rev=1488747526>

Last update: **2017-03-05 21:58**

