

```

1  #!/bin/bash
2  #-----#
3  # Scripts downloaden en installeren. #
4  # #
5  # Geschreven door Karel Zimmer <info@karelzimmer.nl>. #
6  # #
7  # Auteursrecht (c) 2016-2020 Karel Zimmer. #
8  # GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>. #
9  # #
10 # RelNum=26.01.10 #
11 # RelDat=2020-11-16 #
12 #-----#
13
14 #-----#
15 # Global constants #
16 #-----#
17 PROGRAMME=$(basename "$0")
18 if [[ -z $PROGRAMME ]] ||
19    [[ $PROGRAMME = 'bash' ]] ||
20    [[ $PROGRAMME = s* ]]; then
21     readonly PROGRAMME='kzgetscripts'
22 else
23     readonly PROGRAMME
24 fi
25 declare -ir SUCCESS=0
26 readonly CALLED=$(printf '%s' "$0 $*" | xargs --null)
27 readonly LINEH=$(printf '%.0s-' {1..79})
28 readonly STARTDIR=$PWD
29
30 # Terminalattributen ('man terminfo'). Gebruik ${<variabele_naam>}.
31 readonly BOLD=$(tput bold)
32 readonly NORMAL=$(tput sgr0)
33 readonly RED=${BOLD}$(tput setaf 1)
34 readonly GREEN=${BOLD}$(tput setaf 2)
35 readonly BLUE=${BOLD}$(tput setaf 4)
36
37 #-----#
38 # Global variables #
39 #-----#
40 declare TERMINAL=false
41
42 #-----#
43 # Functions #
44 #-----#
45 init_script() {
46 # Enable code-stepping.
47 # trap '(read -p "${BASH_SOURCE:$LINENO} $BASH_COMMAND?")' DEBUG
48     local ubuntu_release
49
50     # Voor Ubuntu 12.04 LTS en Ubuntu 14.04 LTS.
51     ubuntu_release=$(lsb_release --release --short)
52     if [[ $ubuntu_release = '12.04' || $ubuntu_release = '14.04' ]]; then
53         printf '%s\n' "*** Gebruik een andere procedure voor het verkrijgen van \
54 scripts ***"
55
56         1. Installeer de scripts via het starten van een browser en ga naar \
57 karelzimmer.nl.
58         2. Klik op Linux, en onder Scripts, rechtsklik op Scriptspakket en kies Opslaan.
59         3. Ga naar Downloads, klik rechts op kzscripts_365_all.deb en open deze met
60 Software-installatie en klik op Installeren (of klik op Verwijderen en herhaal
61 deze stap).
62         4. Volg de aanwijzingen op het scherm."
63         exit 0
64     fi
65
66     log_debug "Funcname: ${FUNCNAME[0]}"

```

```

67     LOGCMD="systemd-cat --identifier=$PROGNAME --priority=info"
68     LOGCMD_DEBUG="systemd-cat --identifier=$PROGNAME --priority=debug"
69     CHKLOGCMD="journalctl --all --boot --no-pager --identifier=$PROGNAME \
70 --since='$(date '+%Y-%m-%d %H:%M:%S')'"
71     log ">>>> started $PROGNAME as $CALLED (from $STARTDIR)"
72
73     # Script-hardening
74     set -o errexit
75     set -o errtrace
76     set -o nounset
77     set -o pipefail
78     # bash 5.0: trap exit: ${FUNCNAME[0]}: ongebonden variabele
79     trap 'signal ERROR  $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' ERR
80     trap 'signal EXIT   $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' EXIT
81     trap 'signal SIGINT  $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' SIGINT
82     trap 'signal SIGTERM $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' SIGTERM
83     trap 'signal SIGHUP  $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' SIGHUP
84 }
85
86 log() {
87     printf '%b' "$@" |& $LOGCMD
88 }
89
90 log_debug() {
91     printf '%b' "$@" |& $LOGCMD_DEBUG
92 }
93
94 signal() {
95     log_debug "Funcname: ${FUNCNAME[0]}"
96     local signal=${1:-signal?}
97     local -i rc=${2:-1}
98     local -i lineno=${3:-0}
99     local funcname=${4:-funcname?}
100    local apt_repair="Als de pakketbeheerder 'apt' foutmeldingen geeft,
101 start dan een nieuw Terminalvenster, en voer uit:${BLUE}
102 sudo apt-get update --fix-missing
103 sudo dpkg --configure --pending
104 sudo apt-get install --fix-broken
105 sudo update-initramfs -u${NORMAL}"
106    local cmd=${5:-cmd?}
107    local status=''
108    local exit_msg="${RED}Programma $PROGNAME is afgebroken.
109 Controleer de logmeldingen hierboven.${NORMAL}
110 $apt_repair"
111
112    printf "+++++ trap %-7.7s: rc %3d: line %4d: func %s: %-s" \
113           "$signal" "$rc" "$lineno" "$funcname" "$cmd" |& $LOGCMD
114
115    case $signal in
116        ERROR)
117        exit "$rc"
118        ;;
119        EXIT)
120        case $rc in
121            0)
122                status="${GREEN}$rc/OK${NORMAL}"
123                ;;
124            *)
125                status="${RED}$rc/ERROR${NORMAL}"
126                printf '%s\n' \
127                       "$LINEH" \
128                       eval "$CHKLOGCMD" \
129                       printf '%s\n%b\n' \
130                               "$LINEH" \
131                               "$exit_msg"
132                ;;
133    esac

```

```
133         esac
134         log "<<<<< ended.. $status"
135
136         # Een non-gui script gestart met optie gui.
137         if $TERMINAL; then
138             read -rp \
139                 "Druk op de Enter-toets om verder te \
140 gaan [Enter]: " < /dev/tty
141         fi
142
143         trap - ERR EXIT SIGINT SIGPIPE SIGTERM SIGHUP
144         rm --force s s.{1..99}
145         exit "$rc"
146     ;;
147     SIG*)
148         exit "$rc"
149     ;;
150     esac
151 }
152
153 check_input() {
154     log_debug "Funcname: ${FUNCNAME[0]}"
155     local text="$PROGNAME: geen opties of argumenten opgeven\n
156 Voor de nieuwste versie van $PROGNAME én het scriptspakket,
157 gebruik de opdracht: ${BLUE}wget -O- karelzimmer.nl/s | bash${NORMAL}
158 of: ${BLUE}curl -L karelzimmer.nl/s | bash${NORMAL}
159 of: ${BLUE}wget karelzimmer.nl/s; bash s${NORMAL}
160 of: ${BLUE}$PROGNAME${NORMAL}"
161
162     if [[ "$*" ]]; then
163         # Start in grafische modus; Starter: Terminal=true, CLI: optie gui.
164         if [[ "$*" = *-g* ]]; then
165             TERMINAL=true
166         else
167             printf "%s\n" "$text" >&2
168             log_debug "Text: $text"
169             exit $SUCCESS
170         fi
171     fi
172     clear -x
173
174     # Wijzig de terminalemulatie door een bug in Whiptail's Info Box.
175     TERM=ansi
176
177     check_user
178     check_on_ac_power
179 }
180
181 check_user() {
182     log_debug "Funcname: ${FUNCNAME[0]}"
183     local text=''
184
185     text="\n
186 De huidige gebruiker is geen beheerder (is geen lid van groep sudo).\n
187 Dit programma heeft beheerdersrechten nodig, en kan nu niet verder.\n
188 Typ 'exit' om dit venster te sluiten."
189     if ! id --name \
190         --groups \
191         "$USER" \
192         grep --quiet \
193         --word-regexp \
194         --regexp='sudo'; then
195         log_debug "Text: $text"
196         whiptail --backtitle "$PROGNAME" \
197             --title 'Fout' \
198             --infobox "$text" \
```

```

199             13 71
200         exit $SUCCESS
201     fi
202 }
203
204 check_on_ac_power() {
205     log_debug "Funcname: ${FUNCNAME[0]}"
206     local -i on_battery=0
207     local text=""
208
209     text='\n\nDe computer gebruikt nu alleen de accu voor de stroomvoorziening.\n
210 Geadviseerd wordt om de computer aan te sluiten op het stopcontact.'
211     on_ac_power >/dev/null 2>&1 || on_battery=?
212     if [[ on_battery -eq 1 ]]; then
213         log_debug "Text: $text"
214         whiptail --backtitle "$PROGNAME" \
215                 --title 'Waarschuwing' \
216                 --msgbox "$text" \
217                 12 71
218     fi
219 }
220 process_input() {
221     log_debug "Funcname: ${FUNCNAME[0]}"
222     local karel_full_name=''
223     local ubuntu_full_name=''
224
225     install_scriptspackage
226     karel_full_name=$(
227         getent passwd karel |
228         awk -F: '{print $5}' |
229         awk -F, '{print $1}' |
230     )
231     ubuntu_full_name=$(
232         getent passwd ubuntu |
233         awk -F: '{print $5}' |
234         awk -F, '{print $1}' |
235     )
236     if [[ ($USER = karel && $karel_full_name = 'Karel Zimmer') ||
237           ($USER = ubuntu && $ubuntu_full_name = 'Live session user') ]]; then
238         create_development_environment
239     fi
240 }
241
242 install_scriptspackage() {
243     log_debug "Funcname: ${FUNCNAME[0]}"
244     local debname=kzscripts_365_all.deb
245     local current_version=""
246     local installed_version=""
247     local site_deb=https://karelzimmer.nl/apps/kzscripts/$debname
248     local temp_deb=""
249     local text=""
250
251     # Bij Live-sessie de universe repo aanzetten.
252     if [[ $HOSTNAME = 'ubuntu' && $USER = 'ubuntu' ]]; then
253         check_sudo_and_dpkg
254         text='\n\nVoeg pakketbron toe'
255         log_debug "Text: $text"
256         whiptail --backtitle "$PROGNAME" \
257                 --title 'Voortgang' \
258                 --infobox "$text" \
259                 9 23
260         sudo add-apt-repository \
261              --yes \
262              universe |& $LOGCMD
263
264         check_sudo_and_dpkg

```

```
265     text='\n\nVervers pakketlijst'
266     log_debug    "Text: $text"
267     whiptail     --backtitle "$PROGNAME" \
268                 --title     'Voortgang' \
269                 --infobox   "$text" \
270                 9 23
271     sudo        apt-get \
272                 update      |& $LOGCMD
273 fi
274
275 # Verwijder oud pakket scripts.
276 if dpkg --status scripts &> /dev/null; then
277     check_sudo_and_dpkg
278     text='\n\nVerwijder scriptspakket (dit kan even duren)'
279     log_debug    "Text: $text"
280     whiptail     --backtitle "$PROGNAME" \
281                 --title     'Voortgang' \
282                 --infobox   "$text" \
283                 9 48
284     sudo        apt-get \
285                 remove     \
286                 --purge    \
287                 --yes     \
288                 scripts    |& $LOGCMD
289 fi
290
291 # Verwijder pakket kzscripts als versie niet current_version is.
292 if dpkg --status kzscripts &> /dev/null; then
293     installed_version=$(
294         dpkg --list \
295             kzscripts |
296         grep --word-regexp \
297             --regexp='kzscripts' |
298         awk '{print $3}'
299     )
300     current_version=$(echo $debname | awk -F_ '{print $2}')
301     if ! [[ $installed_version = "$current_version" ]]; then
302         check_sudo_and_dpkg
303         text='\n\nVerwijder scriptspakket (dit kan even duren)'
304         log_debug    "Text: $text"
305         whiptail     --backtitle "$PROGNAME" \
306                     --title     'Voortgang' \
307                     --infobox   "$text" \
308                     9 48
309         sudo        apt-get \
310                     remove     \
311                     --purge    \
312                     --yes     \
313                     kzscripts  |& $LOGCMD
314     fi
315 fi
316
317 # Installeer scriptspakket.
318 check_sudo_and_dpkg
319 text='\n\nDownload scriptspakket'
320 log_debug    "Text: $text"
321 whiptail     --backtitle "$PROGNAME" \
322                 --title     'Voortgang' \
323                 --infobox   "$text" \
324                 9 26
325
326 temp_deb=$(mktemp -t "$PROGNAME-XXXXXXXXX.deb")
327 wget --no-verbose \
328     --output-document="$temp_deb" \
329     $site_deb |& $LOGCMD
330
```

```

331     check_sudo_and_dpkg
332     text='\n\nInstalleer scriptspakket (dit kan even duren)'
333     log_debug "Text: $text"
334     whiptail --backtitle "$PROGNAME" \
335             --title 'Voortgang' \
336             --infobox "$text" \
337             9 49
338
339     check_sudo_and_dpkg
340     sudo apt-get \
341          install \
342          --reinstall \
343          --yes \
344          "$temp_deb" |& $LOGCMD
345     rm "$temp_deb" |& $LOGCMD
346 }
347
348 check_sudo_and_dpkg() {
349     log_debug "Funcname: ${FUNCNAME[0]}"
350     local text=''
351     local -i wait_for_aptd=10
352
353     if ! sudo --non-interactive true &> /dev/null; then
354         while true; do
355             text='\nAuthenticatie is vereist voor het
356     installeren van pakket kzscripts\n\nVoer uw wachtwoord in:'
357             log_debug "Text: $text"
358             # Constructie '3>&1 1>&2 2>&3' om stdout en stderr te wisselen.
359             if printf '%s\n' "$(
360                 whiptail --backtitle "$PROGNAME" \
361                         --title 'Authenticatie nodig' \
362                         --ok-button 'Verifiëren' \
363                         --cancel-button 'Annuleren' \
364                         --passwordbox "$text" \
365                         12 37 \
366                         3>&1 1>&2 2>&3
367                 )"
368                 sudo --stdin \
369                     true \
370                     &> /dev/null; then
371                     break
372                 else
373                     text='Geen of verkeerd wachtwoord ingevoerd, of
374     Annuleren gekozen.'
375                     log_debug "Text: $text"
376                     if ! whiptail --backtitle "$PROGNAME" \
377                                 --title 'Fout' \
378                                 --yes-button 'Opnieuw' \
379                                 --no-button 'Stoppen' \
380                                 --yesno "$text" \
381                                 10 45; then
382                         text='\n\nProgramma is gestopt door de gebruiker.'
383                         log_debug "Text: $text"
384                         whiptail --backtitle "$PROGNAME" \
385                                 --title 'Gestopt' \
386                                 --infobox "$text" \
387                                 9 43
388                         exit $SUCCESS
389                     fi
390                 fi
391             done
392         fi
393
394     if ls /snap/core/*/var/cache/debconf/config.dat &> /dev/null; then
395         while sudo fuser \
396             /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \

```

```

397         /var/cache/debconf/config.dat \
398         /snap/core/*/var/cache/debconf/config.dat \
399         &> /dev/null; do
400     text='\n\nWacht totdat andere softwarebeheerders klaar zijn'
401     log_debug "Text: $text"
402     whiptail --backtitle "$PROGNAME" \
403             --title 'Voortgang' \
404             --infobox "$text" \
405             9 53
406     sleep $wait_for_aptd
407 done
408 else
409     while sudo fuser \
410             /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \
411             /var/cache/debconf/config.dat \
412             &> /dev/null; do
413     text='\n\nWacht totdat andere softwarebeheerders klaar zijn'
414     log_debug "Text: $text"
415     whiptail --backtitle "$PROGNAME" \
416             --title 'Voortgang' \
417             --infobox "$text" \
418             9 53
419     sleep $wait_for_aptd
420 done
421 fi
422 }
423
424 create_development_environment() {
425     log_debug "Funcname: ${FUNCNAME[0]}"
426     local ftp_set='set ssl:verify-certificate no'
427     local ftp_from=/httpdocs
428     local ftp_to=$HOME/uploads/karelzimmer.nl/httpdocs
429     local ftp_opts='--delete --verbose'
430     local ftp_cmd="mirror $ftp_opts $ftp_from $ftp_to"
431     local ftp_host=server106.hosting2go.nl
432     local ftp_user=kzimmer
433     local ftp_login=$HOME/.kz-$ftp_host
434     local site_tar=https://karelzimmer.nl/apps/kzscripts/kzscripts.tar.gz
435     local temp_tar=''
436     local text=''
437     local uploads=$HOME/uploads
438     local uploads_old=$HOME/uploads.old
439
440     # Pull scriptsarchief.
441     text='\n\nDownload scriptsarchief'
442     log_debug "Text: $text"
443     whiptail --backtitle "$PROGNAME" \
444             --title 'Voortgang' \
445             --infobox "$text" \
446             9 27
447     temp_tar=$(mktemp -t "$PROGNAME-XXXXXXXXX.tar")
448     wget --no-verbose \
449         --output-document="$temp_tar" \
450         $site_tar |& $LOGCMD
451
452     text='\n\nPak scriptsarchief uit'
453     log_debug "Text: $text"
454     whiptail --backtitle "$PROGNAME" \
455             --title 'Voortgang' \
456             --infobox "$text" \
457             9 26
458
459     # Verwijder eerst alle bestanden uit mappen die in het archief voorkomen.
460     tar --list \
461         --file="$temp_tar" \
462         while IFS= read -r file; do

```

```

463         if    [[ ${file: -1} = / ]]; then
464             rm  --force                               \
465                --recursive                           \
466                "${HOME:?}/$file"*
467         fi
468     done
469
470     # Zet alles wat zonet verwijderd is weer neer uit het archief.
471     tar      --extract                               \
472             --gzip                                  \
473             --directory="$HOME"                    \
474             --file="$temp_tar"                      |& $LOGCMD
475     rm      "$temp_tar"                             |& $LOGCMD
476
477     # Pull website.
478     if ! [[ -f $ftp_login ]]; then
479         text='\nAuthenticatie is vereist voor het\ndownloaden van de website\n
480 Voer het FTP-wachtwoord in:'
481         log_debug "Text: $text"
482         # Constructie '3>&1 1>&2 2>&3' om stdout en stderr te wisselen.
483         printf '%s\n' "$(
484             whiptail  --backtitle    "$PROGNAME"      \
485                      --title       'Authenticatie nodig' \
486                      --ok-button   'Verifiëren'      \
487                      --cancel-button 'Annuleren'     \
488                      --passwordbox "$text"          \
489                      12 37                    \
490                      3>&1 1>&2 2>&3
491         )" > "$ftp_login"
492         chmod 'u=rw,g=,o=' "$ftp_login" |& $LOGCMD
493     fi
494     text='\n\nDownload website (dit kan even duren)'
495     log_debug "Text: $text"
496     whiptail  --backtitle "$PROGNAME" \
497              --title     'Voortgang' \
498              --infobox   "$text" \
499              9 41
500     lftp     --user      "$ftp_user" \
501            --password   "${cat "$ftp_login"}" \
502            -e           "$ftp_set; $ftp_cmd; exit" \
503            "$ftp_host" |& $LOGCMD
504     rsync   --archive \
505            --delete \
506            "$uploads"/ \
507            "$uploads_old" |& $LOGCMD
508 }
509
510 term_script() {
511     log_debug "Funcname: ${FUNCNAME[0]}"
512     local text=''
513
514     text="\n\nVolg nu de stappen zoals beschreven in Checklist installatie.\n
515 Deze Checklist is te vinden op de site https://karelzimmer.nl
516 onder Linux.\n\nTyp 'exit' om dit venster te sluiten."
517     log_debug "Text: $text"
518     whiptail  --backtitle "$PROGNAME" \
519              --title     'Gereed' \
520              --infobox   "$text" \
521              14 65
522     exit $SUCCESS
523 }
524
525
526 #-----#
527 # Main line #
528 #-----#

```

```
529 main() {
530     log_debug "Funcname: ${FUNCNAME[0]}"
531     init_script
532     check_input "$@"
533     process_input
534     term_script
535 }
536
537 main "$@"
538
539 # EOF
```