

```

1  #!/bin/bash
2  # #####
3  # Scripts downloaden en installeren.
4  #
5  # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
6  #
7  # Auteursrecht © 2016-2020 Karel Zimmer.
8  # GNU Algemene Publieke Licentie v3 <https://www.gnu.org/licenses/gpl.html>.
9  # #####
10 REL_NUM=24.01.00
11 REL_DAT=2020-10-08
12 REL_MSG='Extra scherm als Stoppen is gekozen'
13
14 # #####
15 # Settings
16 # #####
17 readonly CALLED=$(printf '%s' "$0 $" | xargs)
18 readonly STARTDIR=$PWD
19 PROGNAME=$(basename "$0")
20 if [[ -z $PROGNAME ]] ||
21    [[ $PROGNAME = 'bash' ]] ||
22    [[ $PROGNAME = s* ]]; then
23     readonly PROGNAME='kzscripts'
24 else
25     readonly PROGNAME
26 fi
27
28 # -----
29 # Global constants
30 # -----
31 declare -ir SUCCESS=0
32 declare -ir ERROR=1
33 declare -ir WARNING=2
34 declare -ir WHIPTAIL_ESC=255
35 readonly DASHES=$(printf '%.0s-' {1..80})
36 readonly DEBNAME=kzscripts_365_all.deb
37 readonly DEBVERSION=$(echo $DEBNAME | awk -F_ '{print $2}')
38 readonly NEW="Voor de nieuwste versie van $PROGNAME én het scriptspakket,
39 gebruik de opdracht: wget -O- karelzimmer.nl/s | bash
40                  of: curl -L karelzimmer.nl/s | bash
41                  of: wget karelzimmer.nl/s; bash s
42                  of: $PROGNAME"
43
44 # Bij aanpassingen ook .completion aanpassen!
45 readonly OPTIONS_SHORT='gghuv'
46 readonly OPTIONS_LONG='debug,gui,help,usage,version'
47 readonly OPTIONS_LONG_SHORT="Lange opties kunnen worden afgekort, zolang de \
48 afkorting uniek blijft."
49 readonly USAGE="Gebruik: $PROGNAME [-d|--debug] [-g|--gui] [-h|--help] \
50 [-u|--usage] [-v|--version] [--]"
51
52 $OPTIONS_LONG_SHORT
53
54 $NEW"
55 readonly HELP="Gebruik: $PROGNAME [OPTIE...]
56
57 Scripts downloaden en installeren.
58
59 Opties:
60     $OPTIONS_LONG_SHORT
61
62     -d --debug          Geef foutopsporingsinformatie weer in het logboek
63     -g --gui            Start in grafische modus
64     -h --help           Deze hulptekst tonen
65     -u --usage          Een korte gebruikssamenvatting tonen
66     -v --version        De versie tonen

```

```

67      --                               Stop verwerken opdrachtregeloptyes
68
69      $NEW"
70      readonly USAGELINE="Typ '$PROGNAME --usage' voor meer informatie."
71
72      # Terminalattributen ('man terminfo'). Gebruik ${<variabele_naam>}.
73      readonly BOLD=$(tput bold)
74      readonly NORMAL=$(tput sgr0)
75      readonly RED=${BOLD}$(tput setaf 1)
76      readonly GREEN=${BOLD}$(tput setaf 2)
77      readonly YELLOW=${BOLD}$(tput setaf 3)
78      readonly BLUE=${BOLD}$(tput setaf 4)
79
80      # -----
81      # Global variables
82      # -----
83      declare CHKLOGCMD=''
84      declare LOGCMD=''
85      declare LOGCMD_DEBUG=''
86      declare OPTION_DEBUG=false
87      declare OPTION_GUI=false
88      declare QUIET=false
89      declare TERMINAL=false
90
91      # #####
92      # Functions
93      # #####
94      check_input() {
95          local -i getopt_rc=0
96          local parsed=''
97
98          set +o errexit
99          parsed=$(
100              getopt --alternative          \
101                     --options              "$OPTIONS_SHORT" \
102                     --longoptions          "$OPTIONS_LONG"  \
103                     --name                 "$PROGNAME"       \
104                     -- "$@"
105          )
106          getopt_rc=$?
107          set -o errexit
108          if [[ $getopt_rc -ne 0 ]]; then
109              printf '%s\n' "$USAGELINE" >&2
110              QUIET=true exit $ERROR
111          fi
112          eval set -- "$parsed"
113
114          while true; do
115              case $1 in
116                  -d|--debug)
117                      debug
118                      shift
119                      ;;
120                  -g|--gui)
121                      OPTION_GUI=true
122                      shift
123                      ;;
124                  -h|--help)
125                      help
126                      QUIET=true exit $SUCCESS
127                      ;;
128                  -u|--usage)
129                      usage
130                      QUIET=true exit $SUCCESS
131                      ;;
132                  -v|--version)

```

```

133         version
134         QUIET=true exit $SUCCESS
135         ;;
136     --)
137         shift
138         break
139         ;;
140     *)
141         shift
142         ;;
143     esac
144 done
145
146 if [[ "$*" ]]; then
147     printf "$PROGNAME: %s\n%s\n" 'geen argumenten opgeven' "$USAGELINE" >&2
148     QUIET=true exit $ERROR
149 fi
150
151 # Als tui-script gestart zoals een gui-script.
152 if $OPTION_GUI; then
153     OPTION_GUI=false
154     TERMINAL=true
155 fi
156
157 clear -x
158 check_user
159 }
160
161 debug() {
162     local text='
163
164 ***** START DEBUG-SESSIE *****'
165
166     whiptail --backtitle "$PROGNAME" \
167             --title 'Debug' \
168             --msgbox "$text" \
169             10 35
170     log_debug '====> START DEBUG-SESSIE'
171     log_debug '----> Start show current environment'
172     log_debug '--- uname --all ---'
173     log_debug "$($uname --all)"
174     log_debug '--- lsb_release --all ---'
175     log_debug "$($lsb_release --all 2> >($LOGCMD_DEBUG))"
176     log_debug '--- cat /etc/os-release ---'
177     log_debug "$($cat /etc/os-release)"
178     log_debug '--- declare -p ---'
179     log_debug "$($declare -p)"
180     log_debug '<---- End show current environment'
181     log_debug '***** Routing bash xtrace to FD4'
182     exec 4> >($LOGCMD_DEBUG)
183     BASH_XTRACEFD=4
184     log '***** Setting bash options'
185     set -o verbose
186     set -o xtrace
187     log_debug '***** Let the fun begin ...'
188     # shellcheck disable=SC2016
189     ps4='+({$BASH_SOURCE}:${LINENO}): ${FUNCNAME[0]:+${FUNCNAME[0}}}(): {'
190     export PS4=$ps4
191     unset ps4
192
193     OPTION_DEBUG=true
194 }
195
196 help() {
197     local manline="Typ 'man $PROGNAME' voor meer informatie."
198

```

```

199     printf '%b\n' "$HELP\n\n$manline"
200 }
201
202 usage() {
203     local helpline="Typ 'help $PROGNAME' voor meer informatie."
204
205     printf '%b\n' "$USAGE\n\n$helpline"
206 }
207
208 version() {
209     local version_info=''
210
211     printf -v version_info          \
212         '%s'                        \
213         "$PROGNAME $REL_NUM ($(date +%d-%b-%Y' \
214         --date=$REL_DAT) - $REL_MSG)
215     Geschreven door Karel Zimmer <info@karelzimmer.nl>.
216
217     Auteursrecht © 2016-$(date +%Y) Karel Zimmer.
218     GNU Algemene Publieke Licentie v3 <https://www.gnu.org/licenses/gpl.html>."
219     printf '%b\n' "$version_info"
220 }
221
222 init_script() {
223     ### ↓↓↓ Ubuntu 14.04 LTS.
224     if command -v systemd &> /dev/null; then
225         LOGCMD="systemd-cat --identifier=$PROGNAME --priority=info"
226         LOGCMD_DEBUG="systemd-cat --identifier=$PROGNAME --priority=debug"
227         CHKLOGCMD="journalctl --all --boot --no-pager \
228         --identifier=$PROGNAME --since='$(date +%Y-%m-%d %H:%M:%S)'"
229     else
230         LOGCMD="logger --tag $PROGNAME --priority info"
231         LOGCMD_DEBUG="logger --tag $PROGNAME --priority debug"
232         CHKLOGCMD="grep $PROGNAME /var/log/syslog"
233     fi
234     ### ↑↑↑ Ubuntu 14.04 LTS.
235     ### ↓↓↓ Ubuntu 16.04 LTS en verder.
236     # LOGCMD="systemd-cat --identifier=$PROGNAME"
237     # LOGCMD_DEBUG="systemd-cat --identifier=$PROGNAME --priority=debug"
238     # CHKLOGCMD="journalctl --all --boot --no-pager --identifier=$PROGNAME \
239     # --since='$(date +%Y-%m-%d %H:%M:%S)'"
240     ### ↑↑↑ Ubuntu 16.04 LTS en verder.
241
242     log ">>>> started $PROGNAME as $CALLED (from $STARTDIR)"
243
244     # Script-hardening
245     set -o errexit
246     set -o errtrace
247     set -o nounset
248     set -o pipefail
249     # bash 5.0: trap exit: ${FUNCNAME[0]}: ongebonden variabele
250     trap 'signal ERROR  $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' ERR
251     trap 'signal EXIT   $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' EXIT
252     trap 'signal SIGINT  $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' SIGINT
253     trap 'signal SIGTERM $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' SIGTERM
254     trap 'signal SIGHUP  $? $LINENO ${FUNCNAME:--} "$BASH_COMMAND"' SIGHUP
255
256     # Wijzig de terminalemulatie door een bug in Whiptail's Info Box.
257     TERM=ansi
258 }
259
260 log() {
261     printf '%b' "$@" |& $LOGCMD
262 }
263
264 log_debug() {

```

```

265 printf '%b' "$@" |& $LOGCMD_DEBUG
266 }
267
268 signal() {
269     local signal=${1:-signal?}
270     local -i rc=${2:-1}
271     local -i lineno=${3:-0}
272     local funcname=${4:-funcname?}
273     local cmd=${5:-cmd?}
274     local apt_repair="
275 Als de pakketbeheerder 'apt' foutmeldingen geeft,
276 start dan een nieuw Terminalvenster en voer uit:${BLUE}
277 sudo apt-get update --fix-missing
278 sudo dpkg --configure --pending
279 sudo apt-get install --fix-broken
280 sudo rm /var/lib/dpkg/lock${NORMAL}"
281     local logtxt=""
282     local status=""
283     local text=""
284
285     ***** EINDE DEBUG-SESSIE *****
286
287     printf -v logtxt "+++++ trap %-7.7s: rc %3d: line %4d: func %s: %-s" \
288         "$signal" "$rc" "$lineno" "$funcname" "$cmd"
289
290     case $signal in
291         ERROR)
292             log "$logtxt"
293             # Systeemvariabele $- bevat alle huidige shell-opties.
294             # Als shell-optie errexit is aangezet, bevat $- een e.
295             # Als uitgezet, geef dan alleen de afsluitwaarde terug.
296             if [[ $- == *e* ]]; then
297                 exit "$rc"
298             else
299                 log "+++++ trap ignored: 'set +r' or 'set +o errexit' \
300 specified"
301                 return "$rc"
302             fi
303             ;;
304         EXIT)
305             log "$logtxt"
306             signal_exit_cleanup
307             if $OPTION_DEBUG; then
308                 set +o verbose
309                 set +o xtrace
310                 BASH_XTRACEFD=""
311                 exec 4>&-
312                 text=""
313
314                 ***** EINDE DEBUG-SESSIE *****
315
316                 whiptail --backtitle "$PROGNAME" \
317                     --title 'Debug' \
318                     --msgbox "$text" \
319                     10 35
320
321                 text=""
322                 Opdracht om de log uit te lezen:
323                 $CHKLOGCMD"
324                 whiptail --backtitle "$PROGNAME" \
325                     --title 'Debug' \
326                     --msgbox "$text" \
327                     10 100
328                 log_debug '==== EINDE DEBUG-SESSIE'
329             fi
330             case $rc in
331                 $SUCCESS)
332                     status="${GREEN}$rc/${NORMAL}"

```

```

331             ;;
332             $WARNING)
333                 status="${YELLOW}$rc/WARNING${NORMAL}"
334             ;;
335             $WHIPTAIL_ESC)
336                 status="${YELLOW}$rc/WHIPTAIL_ESC${NORMAL}"
337             ;;
338             *)
339                 status="${RED}$rc/ERROR${NORMAL}"
340             ;;
341         esac
342         log "<<<<< ended.. $status"
343         if ! $QUIET; then
344             if [[ $rc -eq 0 ]]; then
345                 signal_exit_info
346             elif [[ $rc -eq 2 ]]; then
347                 signal_exit_warning
348             else
349                 signal_exit_error
350             fi
351         fi
352         if $TERMINAL; then
353             read -rp \
354                 "Druk op de Enter-toets om verder te gaan [Enter]: " \
355                 < /dev/tty
356         fi
357         trap - ERR EXIT SIGINT SIGPIPE SIGTERM SIGHUP
358         exit "$rc"
359         ;;
360     SIG*)
361         log "$logtxt"
362         exit "$rc"
363         ;;
364     esac
365 }
366
367 signal_exit_cleanup() {
368     rm --force \
369         s \
370         s.{1..99}
371 }
372
373 signal_exit_error() {
374     local chklogrpt=""
375     local exit_msg="${RED}Programma $PROGNAME is afgebroken.
376 Controleer de logmeldingen hierboven.${NORMAL}$apt_repair"
377     local text=""
378
379     Programma $PROGNAME is afgebroken."
380
381     chklogrpt=$(mktemp -t "$PROGNAME-XXXXXXXXX.log")
382     {
383         printf '%s\n' \
384             "$DASHES"
385         eval "$CHKLOGCMD"
386         printf '%s\n%b\n' \
387             "$DASHES" \
388             "$exit_msg"
389     } >> "$chklogrpt"
390     remove_non_printables "$chklogrpt"
391     whiptail --backtitle "$PROGNAME" \
392             --title 'Fout' \
393             --scrolltext \
394             --textbox "$chklogrpt" \
395             "$((tput lines))" "$((tput cols))"
396     rm "$chklogrpt"

```

```

397     whiptail    --backtitle "$PROGNAME"    \
398                --title      'Afgebroken'    \
399                --infobox    "$text"        \
400                9 41
401 }
402
403 signal_exit_info() {
404     local text=''
405
406     text="
407
408     Volg nu de stappen zoals beschreven in Checklist installatie.
409
410     Deze Checklist is te vinden op de site https://karelzimmer.nl
411     onder Linux."
412     whiptail    --backtitle "$PROGNAME"    \
413                --title      'Gereed'        \
414                --infobox    "$text"        \
415                12 65
416 }
417
418 signal_exit_warning() {
419     local chklogrpt=''
420     local exit_msg="${YELLOW}Programma $PROGNAME is gestopt met een \
421     waarschuwing.
422     Controleer de logmeldingen hierboven.${NORMAL}$apt_repair"
423     local text="
424
425     Programma $PROGNAME is gestopt met een waarschuwing."
426
427     chklogrpt=$(mktemp -t "$PROGNAME-XXXXXXXXX.log")
428     {
429         printf '%s\n'                                \
430             "$DASHES"                                \
431             "$CHKLOGCMD"                              \
432             '%s\n%b\n'                                \
433             "$DASHES"                                \
434             "$exit_msg"                               \
435         } >> "$chklogrpt"
436     remove_non_printables "$chklogrpt"
437     whiptail    --backtitle "$PROGNAME"    \
438                --title      'Waarschuwing'    \
439                --scrolltext \
440                --textbox    "$chklogrpt"      \
441                45 100
442     rm          "$chklogrpt"
443     whiptail    --backtitle "$PROGNAME"    \
444                --title      'Waarschuwing'    \
445                --infobox    "$text"        \
446                9 47
447 }
448
449 remove_non_printables() {
450     local file=${1:-file?}
451
452     sed --in-place --expression='s/[^[:print:]]//g' "$file" # Unprintable chars
453     sed --in-place --expression='s/\[1m//g' "$file" # Bold
454     sed --in-place --expression='s/(B\[m//g' "$file" # Normal 1
455     sed --in-place --expression='s/\[0;10m//g' "$file" # Normal 2
456     sed --in-place --expression='s/\[31m//g' "$file" # Red
457     sed --in-place --expression='s/\[32m//g' "$file" # Green
458     sed --in-place --expression='s/\[33m//g' "$file" # Yellow
459     sed --in-place --expression='s/\[34m//g' "$file" # Blue
460     sed --in-place --expression='s/\[4m//g' "$file" # Underlined
461 }
462

```

```

463 check_user() {
464     local text=""
465
466     De huidige gebruiker is geen beheerder (is geen lid van groep sudo).
467
468     Dit programma heeft beheerdersrechten nodig, en kan nu niet verder."
469
470     if ! id --name \
471         --groups \
472         "$USER" |
473         grep --quiet \
474         --word-regexp \
475         --regexp='sudo'; then
476         whiptail --backtitle "$PROGNAME" \
477             --title 'Fout' \
478             --infobox "$text" \
479             11 71
480         QUIET=true exit $ERROR
481     fi
482 }
483
484 process_input() {
485     install_scriptspackage
486     if [[ $USER = 'karel' || $USER = 'ubuntu' ]]; then
487         create_development_environment
488     fi
489 }
490
491 install_scriptspackage() {
492     local kzscripts_version=""
493     local site_deb=https://karelzimmer.nl/apps/kzscripts/$DEBNAME
494     local temp_deb=""
495     local text=""
496
497     # Bij Live-sessie de universe repo aanzetten.
498     if [[ $HOSTNAME = 'ubuntu' ]] &&
499         [[ $USER = 'ubuntu' ]]; then
500         check_sudo_dpkg_available
501         text="\n\nVoeg pakketbron toe"
502         whiptail --backtitle "$PROGNAME" \
503             --title 'Voortgang' \
504             --infobox "$text" \
505             9 23
506         sudo add-apt-repository \
507             --yes \
508             universe |& $LOGCMD
509
510         check_sudo_dpkg_available
511         text="\n\nVervers pakketlijst"
512         whiptail --backtitle "$PROGNAME" \
513             --title 'Voortgang' \
514             --infobox "$text" \
515             9 23
516         sudo apt-get \
517             update |& $LOGCMD
518     fi
519
520     # Verwijder oud pakket scripts.
521     if dpkg --status scripts && /dev/null; then
522         check_sudo_dpkg_available
523         text="\n\nVerwijder scriptspakket (dit kan even duren)"
524         whiptail --backtitle "$PROGNAME" \
525             --title 'Voortgang' \
526             --infobox "$text" \
527             9 48
528         sudo apt-get \

```



```

529             remove                \
530             --purge                \
531             --yes                   \
532             scripts                 |& $LOGCMD
533     fi
534
535     # Verwijder pakket kzscripts als versie niet DEBVERSION is.
536     if dpkg --status kzscripts &> /dev/null; then
537         kzscripts_version=$(
538             dpkg      --list                \
539                     kzscripts              |
540             grep      --word-regexp        \
541                     --regexp='kzscripts'  |
542             awk       '{print $3}'         |
543         )
544
545         if [[ $kzscripts_version != "$DEBVERSION" ]]; then
546             check_sudo_dpkg_available
547             text="\n\nVerwijder scriptspakket (dit kan even duren)"
548             whiptail --backtitle "$PROGNAME" \
549                     --title 'Voortgang' \
550                     --infobox "$text" \
551                     9 48
552             sudo apt-get                \
553                 remove                 \
554                 --purge                \
555                 --yes                  \
556                 kzscripts              |& $LOGCMD
557         fi
558     fi
559
560     # Pull scriptspakket.
561     text='\n\nDownload scriptspakket'
562     whiptail --backtitle "$PROGNAME" \
563             --title 'Voortgang' \
564             --infobox "$text" \
565             9 26
566     temp_deb=$(mktemp -t "$PROGNAME-XXXXXXXXX.deb")
567     wget --no-verbose \
568         --output-document="$temp_deb" \
569         $site_deb |& $LOGCMD
570
571     check_sudo_dpkg_available
572     text="\n\nInstalleer scriptspakket (dit kan even duren)"
573     whiptail --backtitle "$PROGNAME" \
574             --title 'Voortgang' \
575             --infobox "$text" \
576             9 49
577
578     ### ↓↓↓ Ubuntu 14.04 LTS.
579     check_sudo_dpkg_available
580     set +o errexit
581     sudo dpkg \
582         --install \
583         "$temp_deb" |& $LOGCMD
584     set -o errexit
585     check_sudo_dpkg_available
586     sudo apt-get \
587         --fix-broken \
588         --yes install |& $LOGCMD
589     ### ↑↑↑ Ubuntu 14.04 LTS.
590     ### ↓↓↓ Ubuntu 16.04 LTS en verder.
591     # check_sudo_dpkg_available
592     # sudo apt-get \
593     #     install \
594     #     --reinstall

```

```

595 # --yes \
596 # "$temp_deb" |& $LOGCMD
597 ### ↑↑↑ Ubuntu 16.04 LTS en verder.
598 rm "$temp_deb" |& $LOGCMD
599 }
600
601 check_sudo_dpkg_available() {
602     check_sudo_available
603     check_dpkg_available
604 }
605
606 check_sudo_available() {
607     local text=''
608
609     # shellcheck disable=SC2024
610     if sudo --non-interactive \
611         true \
612         &> /dev/null; then
613         return 0
614     fi
615
616     while true; do
617         text='
618 Authenticatie is vereist voor het
619 installeren van pakket kzscripts
620
621 Voer uw wachtwoord in:'
622         # shellcheck disable=SC2005
623         # Constructie '3>&1 1>&2 2>&3' om stdout en stderr te wisselen.
624         if printf '%s\n' "$(
625             whiptail --backtitle "$PROGNAME" \
626                 --title 'Authenticatie nodig' \
627                 --ok-button 'Verifiëren' \
628                 --cancel-button 'Annuleren' \
629                 --passwordbox "$text" \
630                 12 37 \
631                 3>&1 1>&2 2>&3
632         )" |
633             sudo --stdin \
634                 true \
635                 &> /dev/null; then
636             break
637         else
638             text='
639 Geen of verkeerd wachtwoord ingevoerd, of
640 Annuleren gekozen.'
641             if ! whiptail --backtitle "$PROGNAME" \
642                 --title 'Fout' \
643                 --yes-button 'Opnieuw' \
644                 --no-button 'Stoppen' \
645                 --yesno "$text" \
646                 10 45; then
647                 text=""
648             fi
649             Programma $PROGNAME is gestopt door de gebruiker."
650             whiptail --backtitle "$PROGNAME" \
651                 --title 'Gestopt' \
652                 --infobox "$text" \
653                 9 53
654             QUIET=true exit $WARNING
655         fi
656     fi
657     done
658 }
659
660 check_dpkg_available() {

```

```

661     local -i wait_for_aptd=10
662     local text='
663
664     Wacht totdat andere softwarebeheerders klaar zijn'
665
666     if ls /snap/core/*/var/cache/debconf/config.dat &> /dev/null; then
667         # shellcheck disable=SC2024
668         while sudo fuser \
669             /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \
670             /var/cache/debconf/config.dat \
671             /snap/core/*/var/cache/debconf/config.dat \
672             &> /dev/null; do
673             whiptail --backtitle "$PROGNAME" \
674                 --title 'Voortgang' \
675                 --infobox "$text" \
676                 9 53 \
677                 sleep $wait_for_aptd
678             done
679         else
680             # shellcheck disable=SC2024
681             while sudo fuser \
682                 /var/{lib/{dpkg,apt/lists},cache/apt/archives}/lock \
683                 /var/cache/debconf/config.dat \
684                 &> /dev/null; do
685                 whiptail --backtitle "$PROGNAME" \
686                     --title 'Voortgang' \
687                     --infobox "$text" \
688                     9 53 \
689                     sleep $wait_for_aptd
690                 done
691             fi
692         }
693
694     create_development_environment() {
695         local ftp_set='set ssl:verify-certificate no'
696         local ftp_from=/httpdocs
697         local ftp_to=$HOME/uploads/karelzimmer.nl/httpdocs
698         local ftp_opts='--delete --verbose'
699         local ftp_cmd="mirror $ftp_opts $ftp_from $ftp_to"
700         local ftp_host=server106.hosting2go.nl
701         local ftp_user=kzimmer
702         local ftp_passwd='?'
703         local site_tar=https://karelzimmer.nl/apps/kzscripts/kzscripts.tar.gz
704         local temp_tar=''
705         local text=''
706         local uploads=$HOME/uploads
707         local uploads_old=$HOME/uploads.old
708
709         # Pull scriptsarchief.
710         text='\n\nDownload scriptsarchief'
711         whiptail --backtitle "$PROGNAME" \
712             --title 'Voortgang' \
713             --infobox "$text" \
714             9 27 \
715         temp_tar=$(mktemp -t "$PROGNAME-XXXXXXXXX.tar")
716         wget --no-verbose \
717             --output-document="$temp_tar" \
718             $site_tar |& $LOGCMD
719
720         text='\n\nPak scriptsarchief uit'
721         whiptail --backtitle "$PROGNAME" \
722             --title 'Voortgang' \
723             --infobox "$text" \
724             9 26 \
725
726         # Verwijder eerst alle bestanden uit mappen die in het archief voorkomen.

```

```

727     tar          --list                                \
728                --file="$temp_tar"                    |
729     while        IFS= read -r file; do
730         if        [[ ${file: -1} = / ]]; then
731             rm     --force                                \
732                   --recursive                            \
733                   "${HOME:?}/${file}/*"
734         fi
735     done
736
737     # Zet alles wat zonet verwijderd is weer neer uit het archief.
738     tar          --extract                                \
739                --gzip                                    \
740                --directory="$HOME"                      \
741                --file="$temp_tar"                        |& $LOGCMD
742     rm           "$temp_tar"                             |& $LOGCMD
743
744     # Pull website.
745     text='
746     Authenticatie is vereist voor het
747     downloaden van de website
748
749     Voer het FTP-wachtwoord in:'
750     # shellcheck disable=SC2005
751     # Constructie '3>&1 1>&2 2>&3' om stdout en stderr te wisselen.
752     ftp_passwd="$(
753         whiptail  --backtitle      "$PROGNAME"           \
754                  --title          'Authenticatie nodig'  \
755                  --ok-button      'Verifiëren'           \
756                  --cancel-button  'Annuleren'            \
757                  --passwordbox    "$text"                \
758                  12 37                                \
759                  3>&1 1>&2 2>&3)"
760     text='\n\nDownload website (dit kan even duren)'
761     whiptail  --backtitle "$PROGNAME"                    \
762              --title      'Voortgang'                   \
763              --infobox    "$text"                        \
764              9 41
765     lftp     --user      "$ftp_user"                     \
766             --password   "$ftp_passwd"                   \
767             -e "$ftp_set; ftp_cmd; exit"                 \
768             "$ftp_host"                                   |& $LOGCMD
769     unset    ftp_passwd
770     rsync    --archive   \
771             --delete    \
772             "$uploads"/ \
773             "$uploads_old" |& $LOGCMD
774 }
775
776 term_script() {
777     exit $SUCCESS
778 }
779
780 process_script() {
781     check_input "$@"
782     process_input
783 }
784
785 # #####
786 # Main line
787 # #####
788 main() {
789     init_script
790     process_script "$@"
791     term_script
792 }

```

```
793
794  main "$@"
795
796  # EOF
```