

```
1 #!/bin/bash
2 # shellcheck source=kzcommon.sh
3 #-----#
4 # Scripts controleren. #
5 # #
6 # Geschreven door Karel Zimmer <info@karelzimmer.nl>. #
7 # #
8 # Auteursrecht (c) 2015-2020 Karel Zimmer. #
9 # GNU Algemene Publieke Licentie <https://www.gnu.org/licenses/gpl.html>. #
10 # #
11 # RelNum=11.06.02 #
12 # RelDat=2020-11-16 #
13 #-----#
14
15 #-----#
16 # Global constants #
17 #-----#
18 source "${dirname "$0"}/kzcommon.sh
19 readonly THIS_YEAR=$(date +%Y)
20
21 # Bij aanpassingen ook .completion aanpassen!
22 readonly OPTIONS_SHORT=${OPTIONS_SHORT_COMMON}
23 readonly OPTIONS_LONG=${OPTIONS_LONG_COMMON}
24 readonly USAGE="Gebruik: $PROGNAME $OPTIONS_USAGE_COMMON
25
26 $OPTIONS_LONG_SHORT"
27 readonly HELP="Gebruik: $PROGNAME [OPTIE...]
28
29 Scripts controleren.
30
31 Opties:
32     $OPTIONS_LONG_SHORT
33
34 $OPTIONS_HELP_COMMON"
35
36 #-----#
37 # Globale variabelen #
38 #-----#
39 declare -i MAXRC=0
40
41 #-----#
42 # Functions #
43 #-----#
44 check_input() {
45     local -i getopt_rc=0
46     local parsed=''
47
48     parsed=$(
49         getopt --alternative \
50                --options "$OPTIONS_SHORT" \
51                --longoptions "$OPTIONS_LONG" \
52                --name "$PROGNAME" \
53                -- "$@"
54     ) || getopt_rc=$?
55     if [[ $getopt_rc -ne 0 ]]; then
56         printf '%s\n' "$USAGELINE" >&2
57         QUIET=true exit $ERROR
58     fi
59     eval set -- "$parsed"
60     process_general_options "$@"
61
62     while true; do
63         case $1 in
64             --)
65                 shift
66                 break
```

```
67         ;;
68     *)
69         shift
70         ;;
71     esac
72 done
73
74 if [[ "$*" ]]; then
75     printf "$PROGNAME: %s\n%s\n" 'geen argumenten opgeven' "$USAGELINE" >&2
76     QUIET=true exit $ERROR
77 fi
78
79 # Een non-gui script gestart met optie gui.
80 if $OPTION_GUI; then
81     OPTION_GUI=false
82     TERMINAL=true
83 fi
84
85 check_user
86 }
87
88 process_input() {
89     local script=''
90     local scriptsdir=$HOME/kzscripts
91
92     # Te controleren scripts staan in scriptsdir.
93     cd "$scriptsdir"
94
95     for script in *; do
96         if ! [[ -f $script ]]; then
97             continue
98         fi
99         case $script in
100             CHANGELOG*)
101                 continue
102                 ;;
103             *.1|*.desktop|*.policy)
104                 check_tags
105                 check_lines
106                 check_fields
107                 check_trailing_spaces
108                 ;;
109             *.sh)
110                 check_tags
111                 check_lines
112                 check_fields
113                 check_trailing_spaces
114                 check_shellcheck
115                 ;;
116             *)
117                 check_tags
118                 check_lines
119                 check_fields
120                 check_trailing_spaces
121                 check_shellcheck
122                 check_usage
123                 ;;
124         esac
125     done
126 }
127
128 check_tags() {
129     if grep --quiet \
130         --word-regexp \
131         --regexp='#\X\X\X' \
132         --regexp='#\T\O\D\O' \
```

```

133         --regexp='\#\F\I\X\M\E' \
134         "$script"; then
135         info "
136     ${BOLD}In $script:${NORMAL}
137         Gedit-tags gevonden.
138
139     $(
140         grep      --line-number      \
141                 --word-regexp      \
142                 --regexp='\#\X\X\X' \
143                 --regexp='\#\T\O\D\O' \
144                 --regexp='\#\F\I\X\M\E' \
145                 "$script"          |
146         nl        --number-width=4   \
147                 --number-separator='' \
148                 --body-numbering=n
149     )
150     "
151     fi
152 }
153
154 check_lines() {
155     local -i max_line_length=81
156
157     case $script in
158         *.sh)
159             max_line_length=341
160             ;;
161         *.policy)
162             max_line_length=106
163             ;;
164         *)
165             max_line_length=81
166             ;;
167     esac
168
169     if [[ $(wc --max-line-length "$script" |
170         awk '{print $1}') -gt $max_line_length ]]; then
171         info "
172     ${BOLD}In $script:${NORMAL}
173         Regel is te lang ($(wc --max-line-length "$script" | awk '{print $1}')/\
174     $max_line_length).
175     "
176         MAXRC=$WARNING
177     fi
178
179     if [[ $(wc --lines "$script" | awk '{print $1}') -gt 1000 ]]; then
180         info "
181     ${BOLD}In $script:${NORMAL}
182         Meer dan duizend regels.
183     "
184         MAXRC=$WARNING
185     fi
186 }
187
188 check_fields() {
189     local age=''
190     local checkstring=''
191     local copyright=''
192     local rel_dat=''
193     local releaseday=''
194     local rel_num=''
195     local today=''
196
197     # Prevent finding this string here rather than at the top.
198     checkstring=Auteursrecht

```

```

199     copyright=$(
200         grep      --word-regexp          \
201                 --regexp="$checkstring " \
202                 "$script"                || true
203     )
204     if [[ -z $copyright ]]; then
205         info "
206     ${BOLD}In $script:${NORMAL}
207     Auteursrecht ontbreekt.
208     "
209         MAXRC=$ERROR
210     fi
211     if ! [[ $copyright = *"$THIS_YEAR"* ]]; then
212         info "${BOLD}In $script:${NORMAL}
213     Auteursrechtjaar $THIS_YEAR ontbreekt.
214     "
215         MAXRC=$ERROR
216     fi
217
218     # Prevent finding this string here rather than at the top.
219     checkstring=RelNum
220     rel_num=$(
221         awk -F$checkstring=          \
222             '/'"$checkstring"'/={print $2;exit}' \
223             "$script"
224     )
225     if [[ -z $rel_num ]]; then
226         info "
227     ${BOLD}In $script:${NORMAL}
228     Geen RelNum.
229     "
230         MAXRC=$ERROR
231     fi
232
233     # Prevent finding this string here rather than at the top.
234     checkstring=RelDat
235     rel_dat=$(
236         awk -F$checkstring=          \
237             '/'"$checkstring"'/={print $2;exit}' \
238             "$script"
239     )
240     rel_dat=${rel_dat:0:10}
241     if [[ -z $rel_dat ]]; then
242         info "
243     ${BOLD}In $script:${NORMAL}
244     Geen RelDat.
245     "
246         MAXRC=$ERROR
247     fi
248     if ! date --date="$rel_dat" &> /dev/null; then
249         info "
250     ${BOLD}In $script:${NORMAL}
251     Vrijgavedatum $rel_dat is niet geldig.
252     "
253         MAXRC=$ERROR
254     fi
255     today=$(date +%Y%m%d)
256     releaseday=$(date +%Y%m%d --date="$rel_dat")
257     if [[ $releaseday -gt $today ]]; then
258         info "
259     ${BOLD}In $script:${NORMAL}
260     Vrijgavedatum $rel_dat ligt in de toekomst.
261     "
262         MAXRC=$ERROR
263     fi
264     age=$((today - releaseday))

```

```
265     if [[ $age -gt 10000 ]]; then
266         info "
267     ${BOLD}In $script:${NORMAL}
268     Vrijgavedatum $rel_dat is meer dan een jaar oud.
269     "
270     fi
271
272     # Prevent finding this string here rather than at the end.
273     checkstring=EOF
274     if ! grep --quiet \
275         --word-regexp \
276         --regexp="\ $checkstring" \
277         "$script"; then
278         info "
279     ${BOLD}In $script:${NORMAL}
280     $checkstring ontbreekt.
281     "
282         MAXRC=$ERROR
283     fi
284 }
285
286 check_trailing_spaces() {
287     if grep --quiet \
288         --regexp=' $' \
289         "$script"; then
290         info "
291     ${BOLD}In $script:${NORMAL}
292     Eindspaties gevonden.
293     "
294     $(
295         grep --line-number \
296             --regexp=' $' \
297             "$script" |
298         nl --number-width=4 \
299            --number-separator=' ' \
300            --body-numbering=n
301     )
302     "
303     MAXRC=$ERROR
304     fi
305 }
306 }
307
308 check_shellcheck() {
309     local -i check_rc=0
310
311     shellcheck --external-sources \
312         "$script" || check_rc=$?
313     if [[ $check_rc -gt $MAXRC ]]; then
314         MAXRC=$check_rc
315     fi
316 }
317
318 check_usage() {
319     local -i checkrc=0
320
321     bash "$script" --usage |& $LOGCMD || true
322     checkrc=${PIPESTATUS[0]}
323
324     if [[ $checkrc -gt $MAXRC ]]; then
325         info "
326     ${BOLD}In $script:${NORMAL}
327     $script --usage gaat fout."
328         MAXRC=$ERROR
329     fi
330 }
```

```
331
332 term_script() {
333     QUIET=true exit $MAXRC
334 }
335
336 #-----#
337 # Main line                                     #
338 #-----#
339 main() {
340     init_script
341     check_input "$@"
342     process_input
343     term_script
344 }
345
346 main "$@"
347
348 # EOF
```