

```
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=13.02.00
12: # RelDat=2021-01-18
13: # -----
14: #
15: # -----
16: # Global constants
17: # -----
18: source "$(dirname "$0")"/kzcommon.sh
19: readonly THIS_YEAR=$(date +%Y)
20: readonly LAST_YEAR=$((THIS_YEAR - 1))
21:
22: # Bij aanpassingen ook .completion aanpassen!
23: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
24: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON
25: readonly USAGE="Gebruik: $PROGNAME $OPTIONS_USAGE_COMMON"
26: readonly HELP="Gebruik: $PROGNAME [OPTIE...]"
27:
28: Scripts controleren.
29:
30: Opties:
31: $OPTIONS_HELP_COMMON"
32:
33: # -----
34: # Globale variabelen
35: # -----
36: declare -i MAXRC=0
37:
38: # -----
39: # Functions
40: # -----
41: check_input() {
42:     local -i getopt_rc=0
43:     local parsed=''
44:
45:     parsed=$(
46:         getopt  --alternative           \
47:                 --options               "$OPTIONS_SHORT"         \
48:                 --longoptions          "$OPTIONS_LONG"          \
49:                 --name                  "$PROGNAME"              \
50:                 -- "$@"
51:     ) || getopt_rc=$?
52:     if [[ $getopt_rc -ne 0 ]]; then
53:         printf '%s\n' "$USAGELINE" >&2
54:         exit $ERROR
55:     fi
56:     eval set -- "$parsed"
57:     process_general_options "$@"
58:
59:     while true; do
60:         case $1 in
61:             --)
62:                 shift
63:                 break
64:                 ;;
65:             *)
66:                 shift
67:                 ;;
68:         esac
```

```
69:     done
70:
71:     if [[ "$*" ]]; then
72:         printf "%s\n%s\n" 'geen argumenten opgeven' "$USAGELINE" >&2
73:         exit $ERROR
74:     fi
75:
76:     # Een non-gui script gestart met optie gui.
77:     if $OPTION_GUI; then
78:         OPTION_GUI=false
79:         TERMINAL=true
80:     fi
81:
82:     check_user
83: }
84:
85: process_input() {
86:     local script=''
87:     local scriptsdir=$HOME/kzscripts
88:
89:     # Te controleren scripts staan in scriptsdir.
90:     cd "$scriptsdir"
91:
92:     for script in *; do
93:         if ! [[ -f $script ]]; then
94:             continue
95:         fi
96:         case $script in
97:             CHANGELOG*)
98:                 continue
99:                 ;;
100:            *.1|*.desktop|*.policy)
101:                check_tags
102:                check_lines
103:                check_fields
104:                check_trailing_spaces
105:                ;;
106:            *.completion|*.sh)
107:                check_tags
108:                check_lines
109:                check_fields
110:                check_trailing_spaces
111:                check_script_shellcheck
112:                ;;
113:            *)
114:                check_tags
115:                check_lines
116:                check_fields
117:                check_trailing_spaces
118:                check_script
119:                check_usage
120:                ;;
121:        esac
122:     done
123: }
124:
125: check_tags() {
126:     if grep --quiet \
127:         --word-regexp \
128:         --regexp='#'XXX' \
129:         --regexp='#'TODO' \
130:         --regexp='#'FIXME' \
131:         "$script"; then
132:         info "
133: ${BOLD}In $script:${NORMAL}
134:     Gedit-tags gevonden.
135:
136: $(
```

```
137:     grep      --line-number      \
138:             --word-regexp        \
139:             --regexp='#' 'XXX'    \
140:             --regexp='#' 'TODO'   \
141:             --regexp='#' 'FIXME'  \
142:             "$script"             |
143:     nl         --number-width=4    \
144:             --number-separator=''  \
145:             --body-numbering=n
146: )
147: "
148:     fi
149: }
150:
151: check_lines() {
152:     local -i max_line_length=79
153:
154:     case $script in
155:         *.sh)
156:             max_line_length=465
157:             ;;
158:         *.policy)
159:             max_line_length=106
160:             ;;
161:         *)
162:             max_line_length=79
163:             ;;
164:     esac
165:
166:     if [[ $(wc --max-line-length "$script" |
167:         awk '{print $1}') -gt $max_line_length ]]; then
168:         info "
169: ${BOLD}In $script:${NORMAL}
170: Regel is te lang \
171: $(wc --max-line-length "$script" | awk '{print $1}')/$max_line_length).
172: "
173:         if [[ $WARNING -gt $MAXRC ]]; then
174:             MAXRC=$WARNING
175:         fi
176:     fi
177:
178:     if [[ $(wc --lines "$script" | awk '{print $1}') -gt 1000 ]]; then
179:         info "
180: ${BOLD}In $script:${NORMAL}
181: Meer dan duizend regels.
182: "
183:         if [[ $WARNING -gt $MAXRC ]]; then
184:             MAXRC=$WARNING
185:         fi
186:     fi
187: }
188:
189: check_fields() {
190:     local age=''
191:     local copyright_line=''
192:     local rel_dat=''
193:     local releaseday=''
194:     local rel_num=''
195:     local today=''
196:
197:     copyright_line=$(
198:         grep      --word-regexp      \
199:                 --regexp='#' 'Auteursrecht ' \
200:                 "$script"           || true
201:     )
202:     if [[ -z $copyright_line ]]; then
203:         info "
204: ${BOLD}In $script:${NORMAL}
```

```
205:     Auteursrecht ontbreekt.
206: "
207:     if [[ $ERROR -gt $MAXRC ]]; then
208:         MAXRC=$ERROR
209:     fi
210: fi
211:
212: # Jaarwisseling.
213: if ! grep --quiet --regexp="$THIS_YEAR" "$script"; then
214:     update_fields
215: fi
216:
217: rel_num=$(
218:     awk -F'RelNum''=' \
219:         '/''RelNum''=/{print $2;exit}' \
220:         "$script"
221: )
222: if [[ -z $rel_num ]]; then
223:     info "
224: ${BOLD}In $script:${NORMAL}
225: RelNum ontbreekt.
226: "
227:     if [[ $ERROR -gt $MAXRC ]]; then
228:         MAXRC=$ERROR
229:     fi
230: fi
231:
232: rel_dat=$(
233:     awk -F'RelDat''=' \
234:         '/''RelDat''=/{print $2;exit}' \
235:         "$script"
236: )
237: if [[ -z $rel_dat ]]; then
238:     info "
239: ${BOLD}In $script:${NORMAL}
240: RelDat ontbreekt.
241: "
242:     if [[ $ERROR -gt $MAXRC ]]; then
243:         MAXRC=$ERROR
244:     fi
245: fi
246: if ! date --date="$rel_dat" &> /dev/null; then
247:     info "
248: ${BOLD}In $script:${NORMAL}
249: Vrijgavedatum $rel_dat is niet geldig.
250: "
251:     if [[ $ERROR -gt $MAXRC ]]; then
252:         MAXRC=$ERROR
253:     fi
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:     if [[ $ERROR -gt $MAXRC ]]; then
263:         MAXRC=$ERROR
264:     fi
265: fi
266: age=$((today - releaseday))
267: if [[ $age -gt 10000 ]]; then
268:     info "
269: ${BOLD}In $script:${NORMAL}
270: Vrijgavedatum $rel_dat is meer dan een jaar oud.
271: "
272: fi
```

```
273:
274:     if ! grep      --quiet          \
275:                 --word-regexp      \
276:                 --regexp='# ''EOF' \
277:                 "$script"; then
278:         info "
279: ${BOLD}In $script:${NORMAL}
280:     EOF ontbreekt.
281: "
282:         if [[ $ERROR -gt $MAXRC ]]; then
283:             MAXRC=$ERROR
284:         fi
285:     fi
286: }
287:
288: update_fields() {
289:     local copyright_line_new=''
290:     local copyright_line_first_part=''
291:     local pgm_copyright_line=''
292:     local pgm_copyright_line_new=''
293:
294:     # Man-pagina.
295:     if [[ $copyright_line = .* ]]; then
296:         # get rid of ./" in copyright line.
297:         copyright_line=${copyright_line:4}
298:
299:         # Change program's copyright year in the man page.
300:
301:         # Auteursrecht (c) 2020 Karel Zimmer.
302:         # ->
303:         # Auteursrecht (c) 2020-2021 Karel Zimmer.
304:         pgm_copyright_line="$ $LAST_YEAR Karel Zimmer."
305:         pgm_copyright_line_new="$ $LAST_YEAR-$THIS_YEAR Karel Zimmer."
306:         sed --in-place \
307:             --expression="s/$pgm_copyright_line/$pgm_copyright_line_new/g" \
308:             "$script"
309:
310:         # Auteursrecht (c) 2013-2020 Karel Zimmer.
311:         # ->
312:         # Auteursrecht (c) 2013-2021 Karel Zimmer.
313:         pgm_copyright_line="-$LAST_YEAR Karel Zimmer."
314:         pgm_copyright_line_new="-$THIS_YEAR Karel Zimmer."
315:         sed --in-place \
316:             --expression="s/$pgm_copyright_line/$pgm_copyright_line_new/g" \
317:             "$script"
318:     else
319:         info "
320: ${BOLD}In $script:${NORMAL}
321:     Auteursrechtregel is aangepast voor de jaarwisseling.
322:     Maak zelf een PDF aan. Gebruik script edsha.
323: "
324:     fi
325:     copyright_line_first_part=${copyright_line:0:23}
326:     copyright_line_new=$copyright_line_first_part-$THIS_YEAR Karel Zimmer."
327:     sed --in-place \
328:         --expression="s/$copyright_line/$copyright_line_new/g" \
329:         "$script"
330: }
331:
332: check_trailing_spaces() {
333:     if grep --quiet \
334:         --regexp=' '$' \
335:         "$script"; then
336:         info "
337: ${BOLD}In $script:${NORMAL}
338:     Eindspaties gevonden.
339:
340: $(
```

```
341:         grep      --line-number      \
342:         --regexp=' '$' \
343:         "$script" | \
344:     nl      --number-width=4          \
345:         --number-separator=' ' \
346:         --body-numbering=n
347:
348:     )
349: "
350:     if [[ $ERROR -gt $MAXRC ]]; then
351:         MAXRC=$ERROR
352:     fi
353: fi
354: }
355:
356: check_script() {
357:     if grep --quiet \
358:         --line-regexp \
359:         --regexp='#!'"/bin/bash' \
360:         "$script"; then
361:         check_script_shellcheck
362:     elif grep --quiet \
363:         --line-regexp \
364:         --regexp='#!'"/usr/bin/python3' \
365:         "$script"; then
366:         check_script_pep8
367:     else
368:         info "
369: ${BOLD}In $script:${NORMAL}
370: Onbekend script. Functie check_script() kan script niet controleren."
371:         if [[ $ERROR -gt $MAXRC ]]; then
372:             MAXRC=$ERROR
373:         fi
374:     fi
375: }
376:
377: check_script_shellcheck() {
378:     local -i check_rc=0
379:
380:     shellcheck --external-sources \
381:         "$script" || check_rc=$?
382:     if [[ $check_rc -gt $MAXRC ]]; then
383:         MAXRC=$check_rc
384:     fi
385: }
386:
387: check_script_pep8() {
388:     local -i check_rc=0
389:
390:     pep8 "$script" || check_rc=$?
391:     if [[ $check_rc -gt $MAXRC ]]; then
392:         MAXRC=$check_rc
393:     fi
394: }
395:
396: check_usage() {
397:     local -i checkrc=0
398:
399:     "$scriptsdirec/$script" --usage |& $LOGCMD || check_rc=$?
400:     if [[ $checkrc -gt 0 ]]; then
401:         info "
402: ${BOLD}In $script:${NORMAL}
403: $script --usage gaat fout."
404:         if [[ $ERROR -gt $MAXRC ]]; then
405:             MAXRC=$ERROR
406:         fi
407:     fi
408: }
```

```
409:
410: term_script() {
411:     exit $MAXRC
412: }
413:
414: # -----
415: # Main line
416: # -----
417: main() {
418:     init_script
419:     check_input "$@"
420:     process_input
421:     term_script
422: }
423:
424: main "$@"
425:
426: # EOF
```