

```
1 #!/bin/bash
2 #####
3 # Bestand: ckhtml #
4 # Doel: Valideer HTML- en CSS-bestanden. #
5 # Gebruik: In het terminalvenster: #
6 # ckhtml [OPTIES] #
7 # Auteur: Karel Zimmer (http://karelzimmer.nl, info@karelzimmer.nl) #
8 # ----- #
9 # Auteursrecht © 2012-2017 Karel Zimmer. #
10 # #
11 # Dit programma is vrije software: u mag het herdistribueren en/of wijzigen #
12 # onder de voorwaarden van de GNU Algemene Publieke Licentie zoals #
13 # gepubliceerd door de Free Software Foundation, onder versie 3 van de #
14 # Licentie of (naar uw keuze) elke latere versie. #
15 # #
16 # Dit programma is gedistribueerd in de hoop dat het nuttig zal zijn maar #
17 # ZONDER ENIGE GARANTIE; zelfs zonder de impliciete garanties die #
18 # GEBRUIKELIJK ZIJN IN DE HANDEL of voor BRUIKBAARHEID VOOR EEN SPECIFIEK #
19 # DOEL. Zie de GNU Algemene Publieke Licentie voor meer details. #
20 # #
21 # U hoort een kopie van de GNU Algemene Publieke Licentie te hebben #
22 # ontvangen samen met dit programma. Als dat niet het geval is, zie #
23 # http://www.gnu.org/licenses/. #
24 # ----- #
25 # Versies: 1.0.0 2012-11-25 Eerste versie. #
26 # 2.0.0 2014-09-28 Hernoemd (oude naam is htmlcheck). #
27 # 3.0.0 2015-07-08 Hernoemd (-/- .sh). #
28 # 4.0.0 2016-05-20 Van eigen log naar system log en systemd #
29 # journal. #
30 # 5.0.0 2016-10-22 Hernoemd (oude naam is checkhtml). #
31 #####
32 readonly VERSION_NUMBER=5.6.1
33 readonly RELEASE_DATE=2017-08-22
34
35 #####
36 # Instellingen #
37 #####
38
39 #-----#
40 # Algemene instellingen #
41 #-----#
42 readonly FIRST_COPYRIGHTYEAR=2012 # Eerste auteursrechtjaar
43 readonly PROGDIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)
44 # Programmamap
45 readonly PROGRAMNAME=$(basename "${BASH_SOURCE[0]}")
46 # Programmanaam
47 readonly SCRIPT_NEEDS_SUDO=false # Beheerdersrechten nodig
48 if ! source "$PROGDIR"/script-common.sh; then
49 echo "
50 Het algemeen scriptbestand (script-common.sh) is niet gevonden of bevat \
51 fouten.
52 Is 'wget karelzimmer.nl/s;. s' wel uitgevoerd?
53 Zie voor meer informatie http://karelzimmer.nl, en klik op LEESMIJ." >&2
54 # Return als gesourced (0=bash of -bash).
55 [[ $0 == *bash ]] && return 1 || exit 1
56 fi # Lees alg. variabelen en functies in
57
58 #-----#
59 # Globale constanten #
60 #-----#
61 readonly CSS_DIR=/home/karel/uploads/site/css
62 # Map met CSS
63 readonly HTML_DIR=/home/karel/uploads/site/html
64 # Map met HTML
65
```

```
66 readonly HOME_URL=http://karelzimmer.nl # Home URL
67
68 readonly CSS_URL=$HOME_URL/site/css # URL met CSS
69 readonly HTML_URL=$HOME_URL/site/html # URL met HTML
70
71 readonly CSS_VALIDATOR_URL=http://jigsaw.w3.org/css-validator/validator?uri=
72 # CSS-validator
73 readonly HTML_VALIDATOR_URL=http://validator.w3.org/check?uri=
74 # HTML-validator
75
76 readonly OPT_NEEDS_ARG=false # Optie met een verplicht argument
77 readonly OPTS_SHORT='dhuv' # Korte opties
78 readonly OPTS_LONG='debug,help,usage,version'
79 # Lange opties
80 readonly OPTS_USAGE=$(cat << OPTS_USAGE
81 [-d|--debug] [-h|--help] [-u|--usage] [-v|--version]
82 OPTS_USAGE
83 ) # Gebruiks-opties
84 readonly OPTS_HELP=$(cat << OPTS_HELP
85 -d --debug Geef foutopsporingsinformatie weer in het logboek
86 -h --help Deze hulptekst tonen en stoppen
87 -u --usage Een korte gebruikssamenvatting tonen en stoppen
88 -v --version Scriptversie tonen en stoppen
89 OPTS_HELP
90 ) # Hulp-opties
91 readonly HELP=$(cat << HELP
92 Valideer HTML- en CSS-bestanden.
93
94 Opties:
95 $OPTS_HELP
96
97 Valideer pagina's (.html) en opmaakbestanden (.css) met de W3C Markup- en CSS
98 Validatie Service.
99
100 De namen van de mappen en webadressen staan "hard-coded" in dit script.
101 HELP
102 ) # Hulptekst
103
104 # Foutcodes =====
105 readonly E_DIR_NOT_FOUND=64
106 readonly E_FIREFOX_NOT_STARTED=65
107
108 #-----#
109 # Globale variabelen #
110 #-----#
111
112 # Array, integer =====
113
114 # Boolean =====
115 declare OPT_DEBUG=false # Optie debug opgegeven
116 declare OPT_HELP=false # Optie help opgegeven
117 declare OPT_USAGE=false # Optie usage opgegeven
118 declare OPT_VERSION=false # Optie version opgegeven
119
120 # Tekst =====
121
122 #####
123 # Functies #
124 #####
125
126 #-Functie-----#
127 # Naam: controleer_invoer #
128 # Doel: Controleer en verwerk de invoer van het script. #
129 # Arg.: Script invoer. #
130 #-----#
```

```
131 controleer_invoer() {
132
133     local dir=''
134     local parm_error=false
135
136     # Test op aanwezigheid nieuwe versie getopt.
137     set +o errexit
138     getopt --test > /dev/null
139     getopt_rc=$?
140     set -o errexit
141     if [[ $getopt_rc -eq 4 ]]; then
142
143         # Ontleed (parse) opties en argumenten met getopt.
144         PARSED=$(getopt --alternative          \
145                      --options             $OPTS_SHORT \
146                      --longoptions        $OPTS_LONG  \
147                      --name                "$PROGNAME" \
148                      -- "$@")
149
150         # Controleer op fouten.
151         if [[ $? -gt 0 ]]; then
152             info $HELPLINE
153             exit 2
154         fi
155
156         # Gebruik eval om quoting correct te verwerken.
157         eval set -- "$PARSED"
158     else
159         # Voeg '--' toe aan de invoer i.v.m. eindeloze loop (break op '--').
160         eval set -- "$@ --"
161         log "Oude versie van getopt in gebruik; opties exact opgeven \
162 volgens optie usage (-u)"
163     fi
164
165     # Verwerk de opties en optie-argumenten.
166     while true; do
167         case $1 in
168             -d|--debug)
169                 OPT_DEBUG=true
170                 VERBOSE='--verbose'
171                 shift
172                 ;;
173             -h|--help)
174                 OPT_HELP=true
175                 toon_gebruik $OPT_USAGE
176                 exit 0
177                 ;;
178             -u|--usage)
179                 OPT_USAGE=true
180                 toon_gebruik $OPT_USAGE
181                 exit 0
182                 ;;
183             -v|--version)
184                 OPT_VERSION=true
185                 toon_versie
186                 exit 0
187                 ;;
188             --)
189                 shift
190                 break
191                 ;;
192             *)
193                 error "$PROGNAME: interne fout bij afhandelen optie '$1'"
194                 parm_error=true
195                 shift
```

```
196         ;;
197     esac
198 done
199
200 # Verwerk de non-optie argumenten.
201 if [[ @$@ ]]; then
202     info "$PROGNAME: geen argumenten toegestaan"
203     parm_error=true
204 fi
205
206 # Geef optie help en gebruik aan als er fouten zijn geconstateerd.
207 if $parm_error; then
208     info $HELPLINE
209     exit 3
210 fi
211
212 # Controleer aanwezigheid mappen.
213 for dir in "$CSS_DIR" "$HTML_DIR"; do
214     if [[ ! -d $dir ]]; then
215         error "$PROGNAME: map '$dir' bestaat niet"
216         exit $E_DIR_NOT_FOUND
217     fi
218 done
219
220 # Geef melding om firefox te starten als deze nog niet is gestart.
221 if ! pgrep 'firefox' &> /dev/null; then
222     info "$PROGNAME: Firefox webbrower is niet gestart, start nu \
223 Firefox."
224     read -p 'Druk op Enter-toets wanneer gereed (Enter) '
225     if ! pgrep 'firefox' &> /dev/null; then
226         error 'Firefox webbrower is (nog) niet gestart.'
227         exit $E_FIREFOX_NOT_STARTED
228     fi
229 fi
230
231 return 0
232 }
233
234 #-Functie-----#
235 # Naam: toon_invoer #
236 # Doel: Toon wat het script gaat doen. #
237 # Arg.: Geen argumenten. #
238 #-----#
239 toon_invoer() {
240
241     info " Script: $PROGNAME
242 Valideer:
243 CSS map: $CSS_DIR
244 HTML map: $HTML_DIR
245 "
246
247     return 0
248 }
249
250 #-Functie-----#
251 # Naam: verwerk_invoer #
252 # Doel: Valideer CSS en HTML. #
253 # Arg.: Geen argumenten. #
254 #-----#
255 verwerk_invoer() {
256
257     valideer_css
258     valideer_html
259
260     return 0
```

```
261 }
262
263 #-Functie-----#
264 # Naam: valideer_css #
265 # Doel: Valideer CSS met W3C CSS Validatie Service. #
266 # Arg.: Geen argumenten. #
267 #-----#
268 valideer_css() {
269
270     local filename=''
271     local valurl=''
272
273     cd "$CSS_DIR"
274
275     for filename in *.css; do
276         info "Valideer CSS '$CSS_URL/$filename'."
277         valurl=$CSS_VALIDATOR_URL$CSS_URL/$filename
278         log "$valurl"
279         firefox "$valurl" |& $LOGCMD
280     done
281
282     return 0
283 }
284
285 #-Functie-----#
286 # Naam: valideer_html #
287 # Doel: Valideer HTML met W3C Markup Validatie Service. #
288 # Arg.: Geen argumenten. #
289 #-----#
290 valideer_html() {
291
292     local filename=''
293     local valurl=''
294
295     cd "$HTML_DIR"
296
297     for filename in *.html; do
298         info "Valideer HTML '$HTML_URL/$filename'."
299         valurl=$HTML_VALIDATOR_URL$HTML_URL/$filename
300         log "$valurl"
301         firefox "$valurl" |& $LOGCMD
302     done
303
304     return 0
305 }
306
307 #-Functie-----#
308 # Naam: toon_afsluiten #
309 # Doel: Afsluitende meldingen en/of acties. #
310 # Arg.: Geen argumenten. #
311 #-----#
312 toon_afsluiten() {
313
314     warning 'Controleer de Firefox tabbladen.'
315
316     # Geef "goed"-geluid en (als non-root) toon schermbericht ("bubble").
317     command_exists canberra-gtk-play &&
318         canberra-gtk-play --id=complete |& $LOGCMD
319     command_exists notify-send &&
320         notify-send --urgency=normal \
321             --icon=utilities-terminal \
322             "De $PROGRAMME is uitgevoerd" \
323             "Zie terminalvenster voor meer informatie" |& $LOGCMD
324
325     return 0

```

```
326 }
327
328 #####
329 # Hoofdlijn #
330 #####
331
332 # init_script
333 {
334     controleer_invoer "$@"
335     controleer_gebruiker $SCRIPT_NEEDS_SUDO "$@"
336     toon_invoer
337 }
338
339 # verwerk
340 {
341     toon_gestart
342     verwerk_invoer
343     toon_gestopt
344 }
345
346 # afsl_script
347 {
348     toon_afsluiten
349     exit 0
350 }
351
352 # Einde script.
```