

```
1: #!/bin/bash
2: # shellcheck source=kz-common.sh
3: #####
4: # Webbestanden valideren.
5: #
6: # Geschreven door Karel Zimmer <info@karelzimmer.nl>.
7: #####
8: PROGRAM_PATH=$(realpath "$(dirname "$0")")
9: source "$PROGRAM_PATH"/kz-common.sh
10: PROGRAM_NAME=kz-ckhtml
11: DISPLAY_NAME=${PROGRAM_NAME}/kz-/kz }
12: RELEASE_YEAR=2012
13:
14: VERSION_NUMBER=12.01.02
15: VERSION_DATE=2021-10-04
16:
17:
18: #####
19: # Global constants
20: #####
21:
22: readonly RUN_AS_SUPERUSER=false
23: readonly OPTIONS_SHORT=$OPTIONS_SHORT_COMMON
24: readonly OPTIONS_LONG=$OPTIONS_LONG_COMMON
25: readonly USAGE="Gebruik: $DISPLAY_NAME $OPTIONS_USAGE_COMMON"
26: readonly HELP="Gebruik: $DISPLAY_NAME [OPTIE...]"
27:
28: Webbestanden valideren.
29:
30: Opties:
31: $OPTIONS_HELP_COMMON"
32:
33: readonly CSS_DIR=$HOME/kz-uploads/dist/css
34: readonly HTML_DIR=$HOME/kz-uploads/dist/html
35:
36: readonly HOME_URL=https://karelzimmer.nl
37: readonly CSS_URL=$HOME_URL/css
38: readonly HTML_URL=$HOME_URL/html
39:
40: readonly CSS_VALIDATOR_URL='http://jigsaw.w3.org/css-validator/validator?uri='
41: readonly HTML_VALIDATOR_URL='http://validator.w3.org/check?uri='
42:
43:
44: #####
45: # Global variables
46: #####
47:
48:
49: #####
50: # Functions
51: #####
52:
53: check_input() {
54:     local -i getopt_rc=0
55:     local parsed=''
56:
57:     parsed=$(
58:         getopt --alternative \
59:                --options "$OPTIONS_SHORT" \
60:                --longoptions "$OPTIONS_LONG" \
61:                --name "$DISPLAY_NAME" \
62:                -- "$@"
63:     ) || getopt_rc=$?
64:     if [[ $getopt_rc -ne $SUCCESS ]]; then
65:         printf '%s\n' "$USAGELINE" >&2
66:         exit $ERROR
67:     fi
68:     eval set -- "$parsed"
```

```
69: process_common_options "$@"
70:
71: while true; do
72:     case $1 in
73:         --)
74:             shift
75:             break
76:             ;;
77:         *)
78:             shift
79:             ;;
80:     esac
81: done
82:
83: if [[ "$*" ]]; then
84:     TEXT='geen argumenten opgeven'
85:     printf "$DISPLAY_NAME: %s\n%s\n" "$TEXT" "$USAGELINE" >&2
86:     exit $ERROR
87: fi
88:
89: # Een non-gui script gestart met optie gui.
90: if $OPTION_GUI; then
91:     OPTION_GUI=false
92:     DESKTOP_TERMINAL=false
93: fi
94:
95: check_user
96: }
97:
98:
99: process_input() {
100:     local dir=''
101:
102:     for dir in "$CSS_DIR" "$HTML_DIR"; do
103:         if ! [[ -d $dir ]]; then
104:             error "Map '$dir' bestaat niet."
105:             exit $ERROR
106:         fi
107:     done
108:
109:     if ! pgrep 'chrome' &> /dev/null; then
110:         warning 'Google Chrome webbrowser is nog niet gestart,
111: start nu eerst de Google Chrome webbrowser.'
112:         TEXT="Druk op Enter-toets wanneer Google Chrome webbrowser is gestart \
113: [Enter]: "
114:         read -rp "$TEXT"
115:         if ! pgrep 'chrome' &> /dev/null; then
116:             error 'Google Chrome webbrowser is niet gestart.'
117:             exit $ERROR
118:         fi
119:     fi
120:
121:     validate_css
122:     validate_html
123: }
124:
125:
126: validate_css() {
127:     local filename=''
128:
129:     cd "$CSS_DIR" || exit $ERROR
130:
131:     for filename in *.css; do
132:         google-chrome "$CSS_VALIDATOR_URL$CSS_URL/$filename" |& $LOGCMD
133:     done
134: }
135:
136:
```

```
137: validate_html() {
138:     local filename=''
139:
140:     cd "$HTML_DIR" || exit $ERROR
141:
142:     for filename in *.html; do
143:         google-chrome "$HTML_VALIDATOR_URL$HTML_URL/$filename" |& $LOGCMD
144:     done
145: }
146:
147:
148: term_script() {
149:     info 'Controleer de tabbladen in Google Chrome.'
150:     exit $SUCCESS
151: }
152:
153:
154: #####
155: # Main line
156: #####
157:
158: main() {
159:     init_script "$@"
160:     check_input "$@"
161:     process_input
162:     term_script
163: }
164:
165:
166: main "$@"
167:
168:
169: # EOF
```