

Aufgabe für den Beleg im Fach Internet-Technologien I

Entwickeln Sie eine Progressive Web App (PWA). Die PWA soll einen Countdown bis zu einer bestimmten Uhrzeit an einem bestimmten Datum anzeigen.

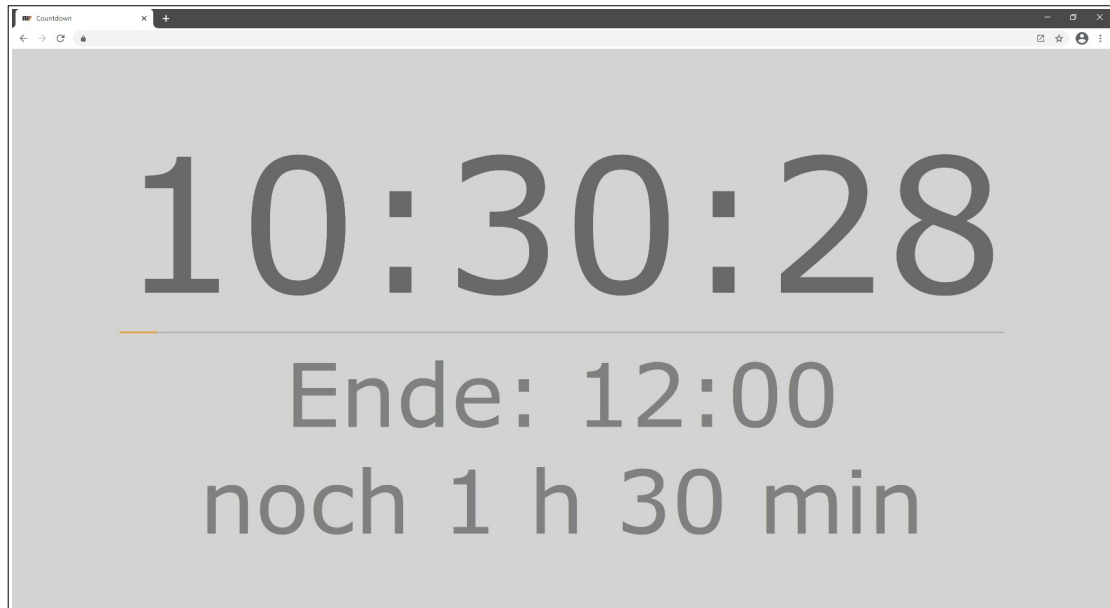


Bild 1: Countdown-PWA im Webbrowser

(1) Grundfunktion

Es sollen folgende Informationen angezeigt werden:

- die aktuelle Uhrzeit mit Stunden-, Minuten- und Sekundenanzeige (sekündlich aktualisiert)
- die Endzeit
- die noch verbleibende Zeit.

Die verbleibende Zeit soll immer in Stunden und Minuten angegeben werden, egal wie weit das Zieldatum in der Zukunft liegt. Ab dem Wert 60 sollen nur noch Minuten angezeigt werden (also nicht „1 h 00 min“ sondern „60 min“ usw.). Die verbleibende Zeit soll einen Offset von 1 Minute erhalten, so dass die Minute, die gerade heruntergezählt wird in der verbleibenden Zeit als voll dargestellt wird (z.B.: „noch 1 min“ obwohl gerade noch 33 s übrig sind). Wenn Stunden vorhanden sind sollen die Minuten immer zweistellig ausgegeben werden, also ggf. mit führender Null.

(2) Input mit Voreinstellungen, Topbar

Das End-Datum und die End-Uhrzeit sollen in entsprechende Input-Felder für date und time eingegeben werden können. Als Voreinstellung soll im time-input [\[8\]](#) die übernächste volle Stunde stehen, im date-input [\[7\]](#) das dazu passende Datum (Sonderfall ab 2 h vor Mitternacht beachten, hier am Besten mit Millisekunden arbeiten). Diese Einstellungsmöglichkeiten sollen sich in einem Topbar [\[9\]](#) befinden, welches bei Bedarf eingeblendet wird.

(3) Speicherung, Auslesen und Reset der End-Werte

End-Datum und End-Uhrzeit sollen auf dem Endgerät gespeichert werden, so dass die PWA auch nach einem Beenden und Wiederstarten auf diesem Endgerät mit den zuletzt eingegebenen Werten weiterläuft. Die Speicherung soll per localStorage-Property(s) [\[10\]](#) geschehen. Zusätzlich sollen diese Werte im laufenden Betrieb zurückgesetzt werden können.

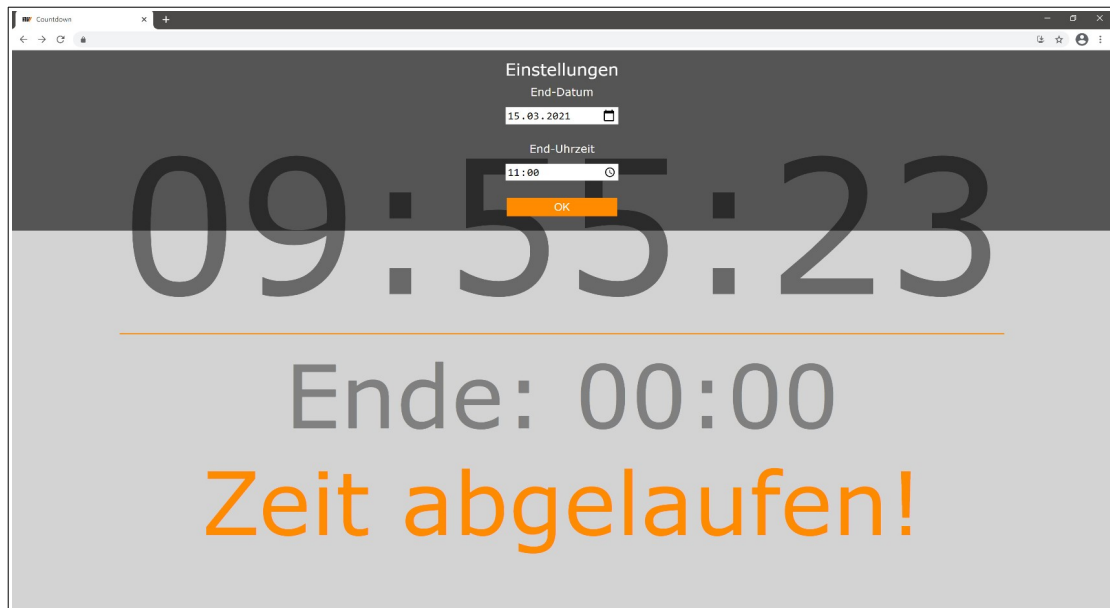


Bild 2: Topbar für Eingaben

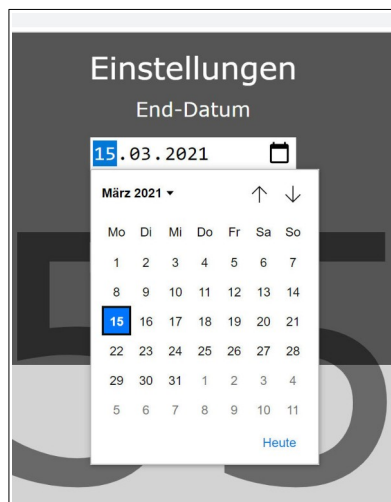


Bild 3: date-input



Bild 4: time-input

(4) Fortschrittsbalken

Die PWA soll einen Fortschrittsbalken [11] besitzen. Beim Start des Countdowns soll er leer sein und sich mit fortschreitender Zeit prozentual farblich füllen.

(5) Benachrichtigung per Farbwechsel und Blinken

Jede viertel, halbe, dreiviertel und volle verbleibende Stunde soll die Anzeige der verbleibenden Zeit für eine Minute die Farbe wechseln und blinken. Das selbe soll bei den letzten 10, 5 und der allerletzten Minute erfolgen. Nach Ende des Countdowns soll die Ausschrift: „Zeit abgelaufen“, ebenfalls blinkend, ausgegeben werden.

(6) Responsivität

Die PWA soll responsiv sein. Je nach verfügbarer Breite sollen die Ausgaben skaliert werden. Selbiges gilt für das Topbar für die Einstellungen. Hier sollen die Eingabeelemente so skaliert werden, dass die PWA auf mobilen Geräten im Hochformat bedienbar bleibt [13].

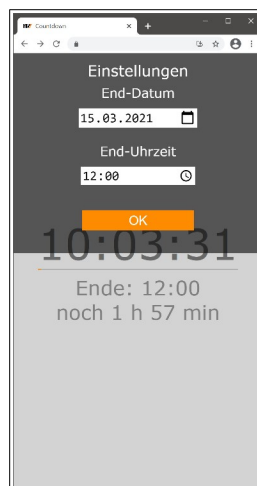


Bild 5: PWA im Hochformat

(7) PWA-Installation

Die PWA soll sich auf Endgeräten installieren lassen. Dafür ist ein Manifest mit Icons [5] und ein Service Worker [12] anzulegen. Zur Installation muss die PWA per https auf einem Webserver mit gültigem offiziellem Zertifikat (z.B. www.htw-dresden.de) oder auf einem lokalen Webserver (z.B. Add-On „Web Server for Chrome“) erreichbar sein.

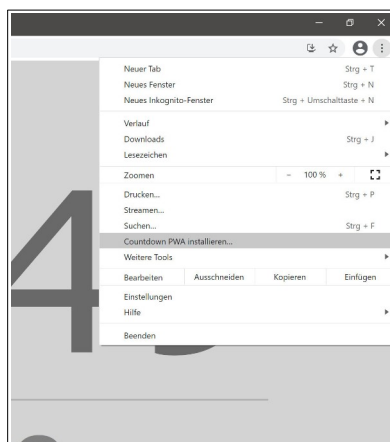


Bild 6: PWA-Installation unter Chrome

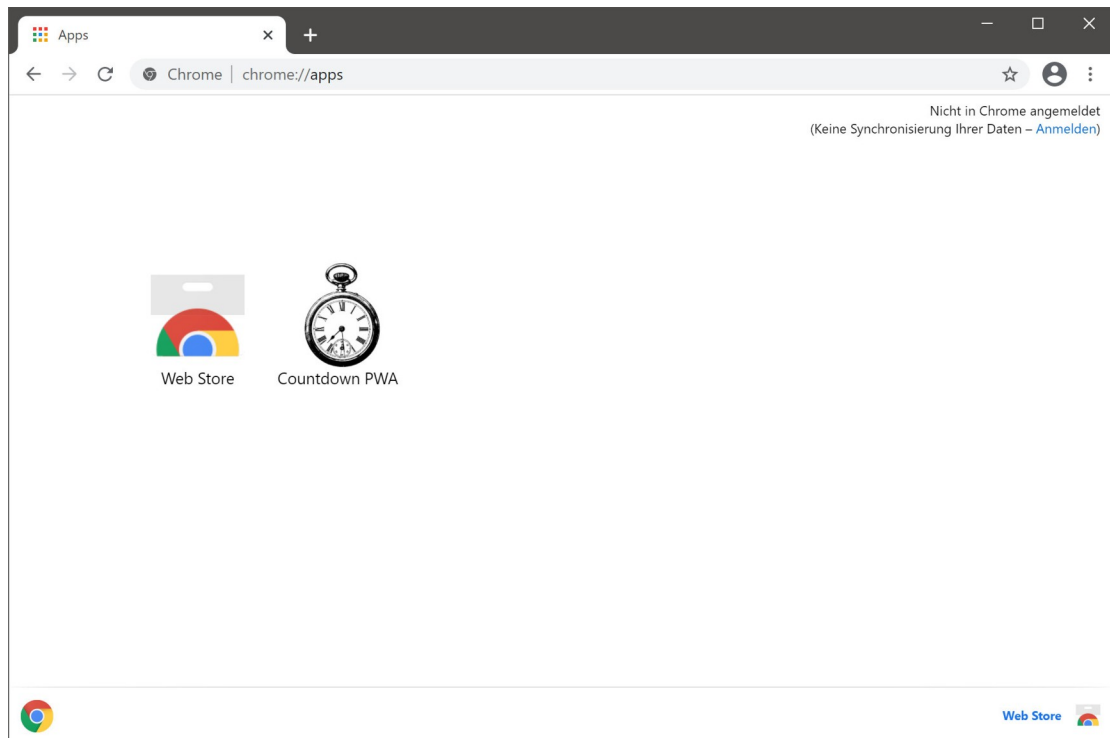


Bild 7: Installierte PWA unter Chrome-Apps

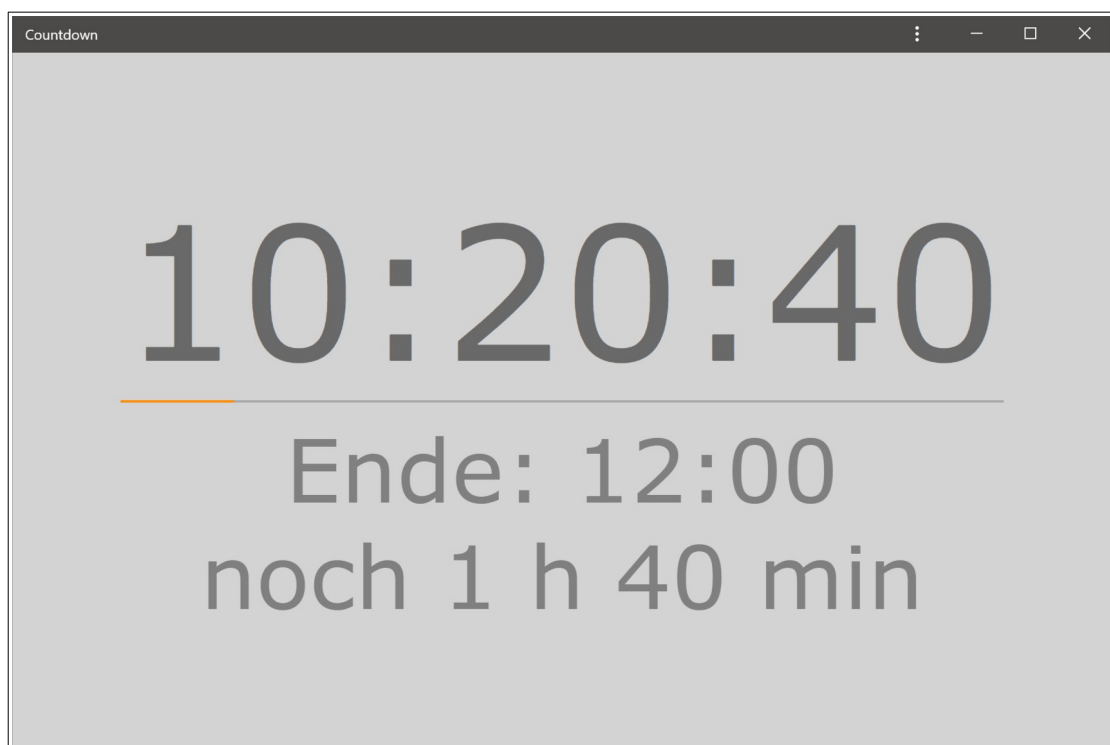


Bild 8: Installierte PWA auf PC ausgeführt

(8) Push-Notifications

Die PWA soll in der Lage sein, zu den in [\[5\]](#) definierten Zeiten Push-Notifications [\[2\]](#) zu versenden.

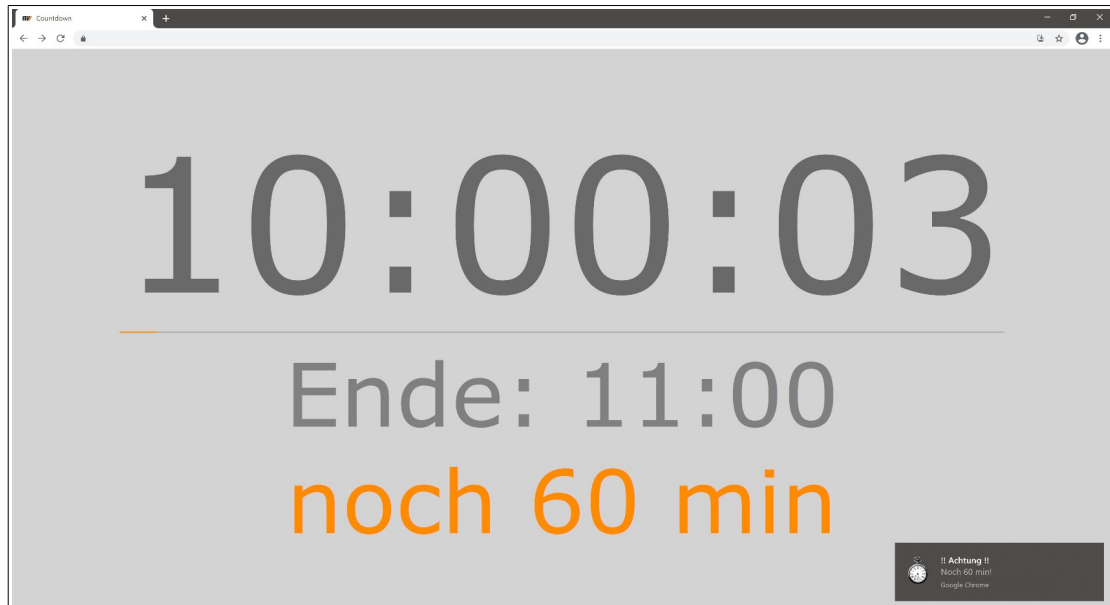


Bild 9: Push-Notification alle 15 min

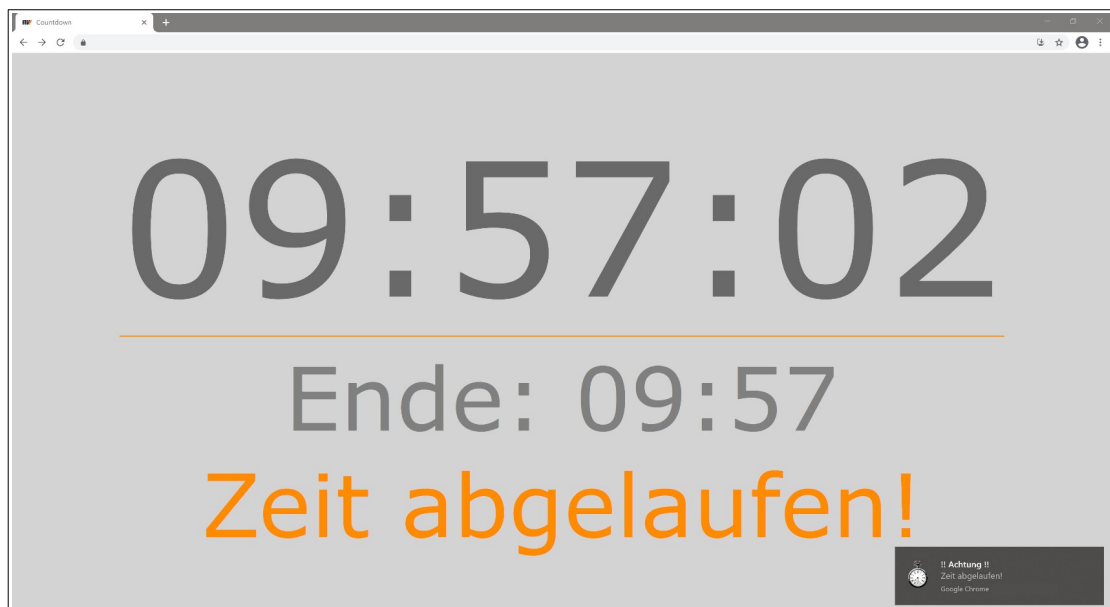


Bild 10: Push-Notification am Ende

(9) Offline-Fähigkeit

Die PWA soll auch benutzbar sein, wenn zeitweise keine Internetverbindung besteht. Erweitern Sie dafür den Service Worker. Überprüfen Sie die Offline-Fähigkeit mit Hilfe der Chrome-Entwicklertools. Setzen Sie dafür unter „Application“ - „Service Workers“ den Haken bei „Offline“ und laden Sie die PWA neu. Die PWA muss jetzt aus dem Cache des Service Workers geladen werden und weiterhin bedienbar sein.

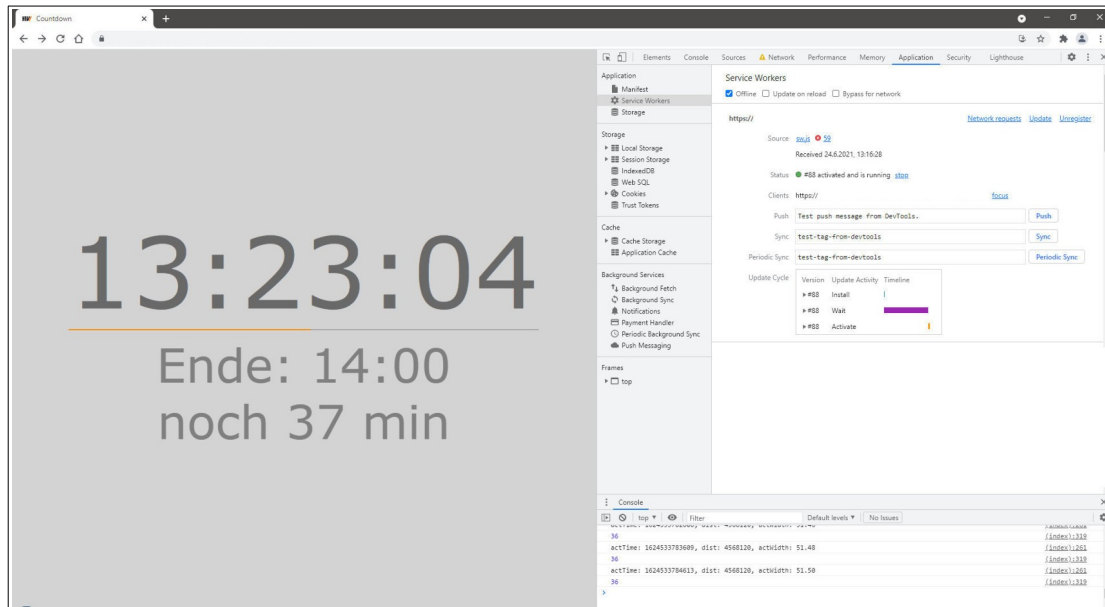


Bild 11: Test der Offline-Funktionalität

Lösen Sie die Aufgabenstellung unter Verwendung von HTML5, Javascript und CSS. Die Nutzung von Frameworks wie Angular, React, Vue u.ä. ist nicht erlaubt. Ihr Design kann von den Screenshots abweichen, die geforderten Informationen müssen aber dargestellt werden.

Nutzen Sie zur Entwicklung und zum Testen Ihrer PWA den Google Chrome Webbrowser und dessen Entwicklertools. Installieren Sie das Add-On „Web Server for Chrome“ als lokalen Webserver. Die Abnahme und Bewertung Ihres Beleges erfolgt unter dieser Umgebung.

Fangen Sie, wenn nötig, leere Eingaben und Fehlerzustände ab.

Schreiben Sie verständlichen, strukturierten Code und kommentieren Sie diesen sinnvoll.

Der Beleg ist ein Einzelbeleg. Plagiate werden mit der Note 5 bewertet.

Die Abgabe des Beleges hat bis spätestens zum **15.01.2023, 23:59 Uhr** zu erfolgen. Packen Sie dazu Ihre Dateien in ein Zip-Archiv und senden Sie dieses per Email an:

paul@informatik.htw-dresden.de .

Vergessen Sie bitte nicht, in der Email Ihren **Studiengang** sowie Ihren **Namen** und Ihre **S-Nummer** anzugeben.

Fragen zum Beleg können Sie jederzeit an o.g. Email-Adresse richten.

Benotungsschema

Genügend:

- [\(1\)](#) Grundfunktion
- [\(2\)](#) Input ohne Voreinstellungen, ohne Topbar, per Menü o.ä.

Befriedigend:

+

- [\(2\)](#) Input mit Voreinstellungen, Topbar
- [\(3\)](#) Speicherung, Auslesen und Reset der Endwerte

Gut:

+

- [\(4\)](#) Fortschrittsbalken
- [\(5\)](#) Benachrichtigung per Farbwechsel und Blinken
- [\(6\)](#) Responsivität

Sehr gut:

+

- [\(7\)](#) PWA-Installation
- [\(8\)](#) Push-Notifications
- [\(9\)](#) Offline-Fähigkeit

Nützliche Links

- [1]
<https://www.sitepoint.com/build-javascript-countdown-timer-no-dependencies/>
- [2]
<https://developers.google.com/web/ilt/pwa/introduction-to-push-notifications>
- [5]
<https://www.favicon-generator.org/>
- [6]
<https://html-online.com/articles/blinking-text-css-animation/>
- [7]
https://www.w3schools.com/tags/att_input_type_date.asp
- [8]
https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_input_type_time
- [9]
https://www.w3schools.com/howto/howto_js_collapse_sidebar.asp
- [10]
https://www.w3schools.com/jsref/prop_win_localstorage.asp
- [11]
https://www.w3schools.com/w3css/w3css_progressbar.asp
- [12] minimaler Serviceworker
<https://www2.htw-dresden.de/~paul/mathe-demo/>
- [13]
https://www.w3schools.com/css/css_rwd_mediaqueries.asp