

Javascript pointed task

12/1/2016

Time: 90 minutes (you have to <i>send</i> solution <i>before</i> 11:50) Points: 25

1 Sending the solution

Pack your files (.js and .css) into zip archive and name it login.zip, substitute login with your login in faculty network.

Send via email:

To: jan.karowski@mini.pw.edu.pl

Subject: [WebApp] JS

(substitute login with your login in faculty network)

Attachments: .zip

Content: empty

2 The task: countdown timer

You have to create a simple web application which will provide user with countdown timers. Each countdown timer consists of an input field for expiration time, delete button and the text displaying expiration date and remaining time.

There are screenshots of the app attached.

You are provided with a timers.html file. The file already contains elements which refer to external stylesheet and javascript file. JQuery is also included. You must not modify this file while solving the task. You have to create js and css files with names used in provided html file and put all your code into those files.

The app does not have to handle occurrence of midnight in any way (e.g. by restarting timers that expired). This behavior is unspecified in task.

2.1 General requirements

You have to send only your .css and .js files (no html, no jquery). Your solution will be checked against the original timers.html and has to work with it. If your solution will not work with the original html, it will be considered incorrect.

Creation of global variables in javascript code is strictly forbidden. **Using global variables results in 5 penalty points.** Starting from requirement 3 all elements have to work separately for each timer created. If some functionality work only for one timer number of points for it is halved.

2.2 Detailed Requirements

- The app contains "Add new timer" button after loading (Figure 1)
- After clicking the button a new timer is created. The timer contains the following elements (Figure 2):
 - input field for expiration time (text input)

- delete button
- html canvas element of size 100×100 under the input

Any number (within browser's resource limits) of timers can be created.

3. 2 points After clicking delete button the timer containing this button is removed.
4. 2 points A counter for created timers is present next to "Add new timer" button. The counter is increased after every creation of a timer. The counter is **not** decreased upon deletion of a timer.
5. 3 points After changing content of input field in timer the text is parsed. The input format for the input field is **HH:MM:SS**. If the format is correct (two digits for hour, two digits for minutes, two digits for seconds and values are correct (**HH** is between 00 and 23 and **MM**, **SS** between 00 and 59)) then a text describing expiration time appears under the input. If there was expiration time before, then it is replaced otherwise nothing happens.
6. 5 points When correct expiration time is provided a countdown timer appears, showing the time remaining to expiration in **HH:MM:SS** format updated every second (Figure 3). When expiration time is changed then countdown is updated:
 - if it is incorrect, then countdown timer disappears, see Figure5
 - if it is correct then new expiration time is set and countdown is updated properly

When expiration time is reached, then countdown timer shows the word "expired".
7. 2 points Frame around not expired timer is green and around the expired one is red (Figure 4). Changing expiration time to new time in future changes expired frame to green again.
8. 6 points When a correct time is provided in input window a circle is drawn on a canvas under the input. When timer has expired the circle is red, when not – green. The circle is animated – it moves on arc shape around the canvas moving 6 degrees each second – imagine a clock hand showing seconds and a circle attached to end of it. See Figure 6 for movement explanation. When incorrect time is provided, the circle does not move. A video file containing circle movement is also provided in `timer.mp4`.

Timers

Add new timer

Timer counter: 0

Figure 1: Initial web page containing "Add new timer" button

Timers

Add new timer

Timer counter: 2

The screenshot shows two identical timer entries stacked vertically. Each entry is enclosed in a green rectangular border. The top entry contains the text "End:" followed by an input field with the placeholder "HH:MM:SS" and a "delete" button to its right. The bottom entry is identical to the top one.

Figure 2: After adding two timers

Timers

Add new timer

Timer counter: 2

The screenshot displays two timer entries, each enclosed in a green rectangular border. The top timer shows an 'End' time of 23:40:22, a 'delete' button, and a status of 'Expires on: 23:40, counting:0:41:36'. A green dot is positioned in the lower-left area of this timer's box. The bottom timer shows an 'End' time of 23:35:50, a 'delete' button, and a status of 'Expires on: 23:35, counting:0:37:4'. A green dot is positioned in the upper-left area of this timer's box.

Figure 3: Timer running after supplying valid expiration time.

Timers

Add new timer

Timer counter: 2



Figure 4: Timer after expiration.

Timers

Add new timer

Timer counter: 2

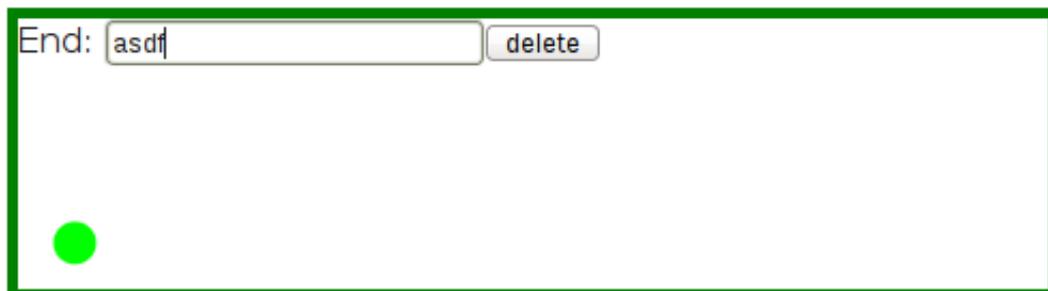


Figure 5: Timer state after changing expiration time to illegal value when timer running.

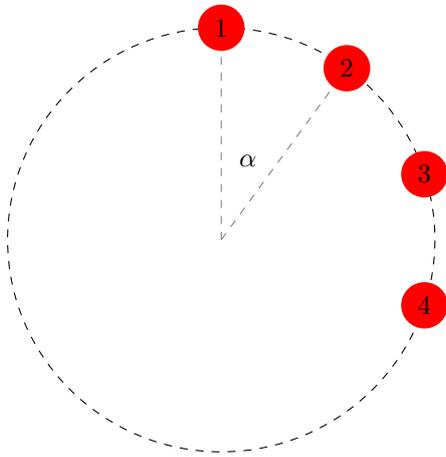


Figure 6: Movement of a circle on a canvas element. Dotted line shows the circle route (and is not drawn in the application). α angle is increased by 6 degrees every second.