Coding Challenge

OVERVIEW

Given a set of events, each with a start time and end time, render the events on a single day calendar (similar to Outlook, Calendar.app, and Google Calendar). There are several properties of the layout:

- No events may visually overlap.
- If two events collide in time, they MUST have the same width. This is an invariant. Call this width W.
- W should be the maximum value possible without breaking the previous invariant.
- Each event is represented by a JavaScript object with a start and end attribute.
- The value of these attributes is the number of minutes since 9am. So {start:30, end:90) represents an event from 9:30am to 10:30am.
- The events should be rendered in a container that is 620px wide (600px + 10px padding on the left/right) and 720px long (the day will end at 9pm).
- The styling of the events should match the attached screenshot (page 3).

CODE GUIDELINES

You may structure your code however you like, but you must implement the following function in the global namespace. The function takes in an array of events and will lay out the events according to the above description.

function layOutDay (events) { }

This function will be invoked from the console for testing purposes. If it cannot be invoked, the submission will be rejected. This function should be idempotent.

In your submission, please implement the calendar with the following input:
[{start: 30, end: 150}, {start: 540, end: 600}, {start: 560, end: 620}, {start: 610, end: 670}];

FAQ

Are frameworks such as JQuery, React, etc. allowed? No. Please use only pure JavaScript, CSS, and native browser APIs only.

Is there a maximum bound on the number of events? You can assume a maximum of 100 events for rendering reasons, but your solution should be generalized.

Do I have to guard against invalid input? You may assume valid input.

What browsers need to be supported? Your solution should work in all newest standards-compliant browsers - please use this as an opportunity to show your knowledge of the web's newest features.

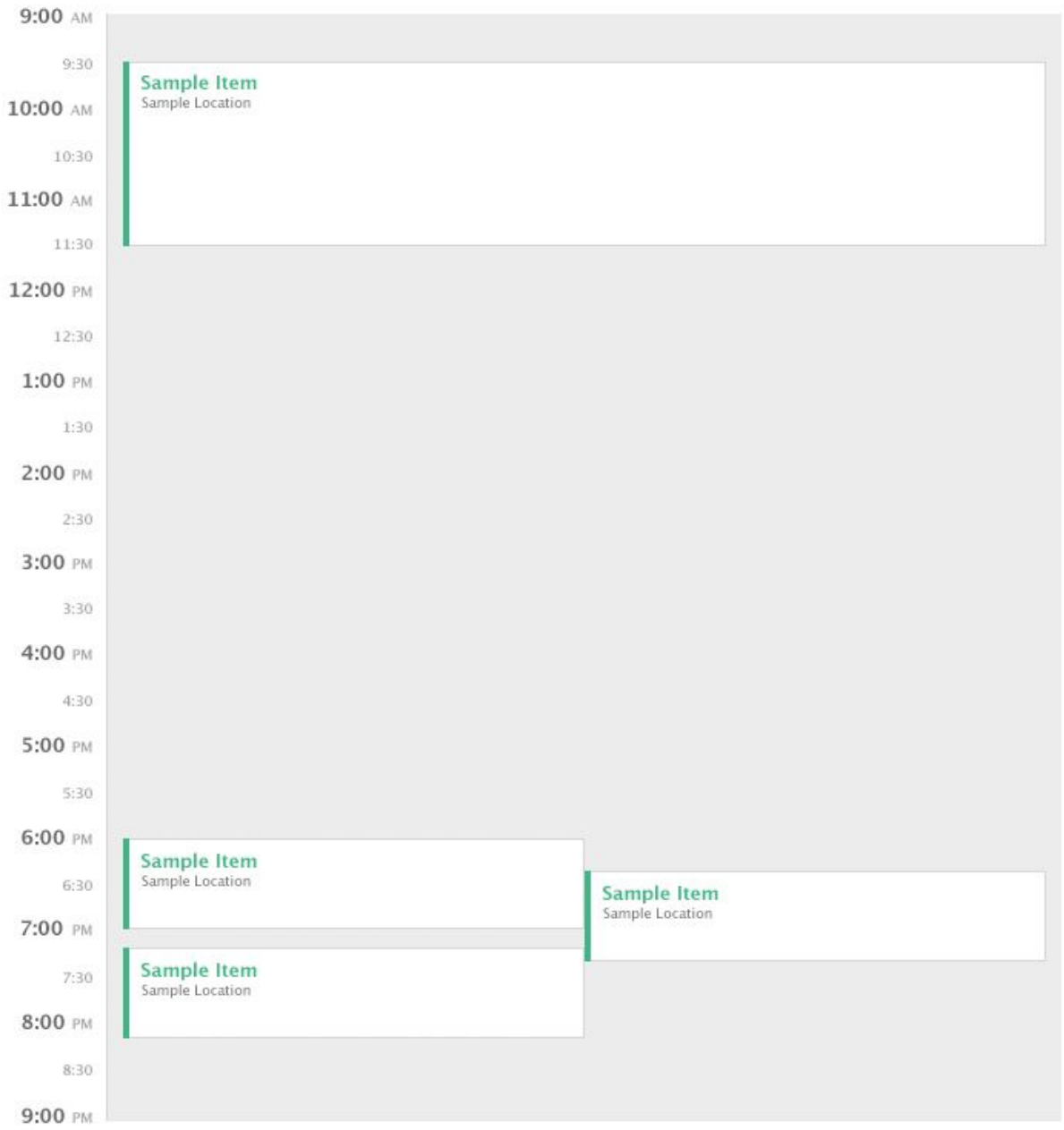
Does my solution need to match the image pixel for pixel? No, we will not be testing for pixel matching.

How will you be testing my solution? We will be running tests from the browser console by invoking the layOutDay() function. Your solution should not require a local web server (e.g. run from localhost) or have any other dependencies besides your html/css/js.

May I ask for help from my colleagues and friends? Please feel free to reach to me with any questions (stefan.ritter@sap.com), but do not share this coding challenge with anyone else as we would like to keep it confidential to keep the interview process fair for all.

Every time layOutDay() is called, does it reset the calendar (clear out the events currently in the calendar first) before adding the new events passed in the function called? I assume it resets because the function is "idempotent"? Every time layOutDay is called it should completely reset everything and start over from scratch

SCREENSHOT


9:00 AM
9:30
Sample Item
Sample Location
10:00 AM
10:30
11:00 AM
11:30
12:00 PM
12:30
1:00 PM
1:30
2:00 PM
2:30
3:00 PM
3:30
4:00 PM
4:30
5:00 PM
5:30
6:00 PM
Sample Item
Sample Location
6:30
Sample Item
Sample Location
7:00 PM
Sample Item
Sample Location
7:30
8:00 PM
8:30
9:00 PM