

Calendar Challenge

This challenge is to render a series of events on a single day calendar, similar to what you'll find on your favourite calendar app.

Your solution should accept as input an `Array` of `events`. An `event` is a JavaScript Object that contains a `start` time and an `end` time as follows:

```
const events = [{start: 30, end: 120}, {start: 300, end: 330}, {start: 290, end: 330}]
```

The `start` and `end` are minutes since 9am, so 30 is 9:30 am, 120 is 11am, etc. Your calendar view is expected to only show the working hours window of 9am-6pm for one day.

Rules:

- The calendar is to be laid out vertically so that later events appear further down the page.
- No events should visually overlap on the calendar.
- Events should make maximum use of the horizontal space. An event with no conflicts should render at full width, two colliding events should both render at half width, etc.
- You must include a function called `renderDay` that is accessible globally on the browser `window` object. It should take 1 argument - `events` - an array of `Event`s as described above. Running `window.renderDay([{start: 30, end: 120}, {start: 300, end: 330}, {start: 290, end: 330}])` in the browser console should render 3 events in your calendar (this should clear any previous events and only render the new ones). **If this function is not present, we will be unable to test your solution.**
- Use of CSS Grid or HTML Tables is not allowed for the rendering of events - this challenge is meant to primarily showcase your JS ability.
- You may use any third party libraries or frameworks that you deem appropriate, but you must provide the core of the implementation.

Submission Format:

Your solution must work without a back-end server. It should have no dependencies other than HTML, CSS and JS. This means your application must work correctly when we open your HTML file in a browser.

- If your application does not require a compile step, your index HTML file will be considered the application entry point.
- If you build your CSS/JS through a compiler, please include both the uncompiled source and the output folder in your submission. We will open the index HTML file from your output folder.
- For security reasons we will not be able to run `npm install`, `npm build` or similar, so it is **very important** that your output folder is included and you have given instructions of how to open your solution (e.g. the location of the index.html file to open). Don't forget to run your build step after you make the final change to your source!
- If you use a compiler, please include your `package.json` file, but *do not* include your `node_modules` folder.

Suggestions:

- This challenge will be the first time we get to see your technical skills. We suggest you use the tools you are most experienced with.
- You should write production quality code, providing tests and comments where necessary. Even though this is just a coding challenge, treat it in the same way you would a core feature of a product.
- Start early and make sure you give yourself enough time to try a different approach if you get stuck.

Pre-Submission Checklist:

- ☐ Your solution works on modern versions of Chrome and Firefox.
- ☐ There are no references to your name or anything that personally identifies you in any file or filename. This is to preserve the anonymity of the assessment. If you have a `.git` folder, delete it.
- ☐ If there are any steps required to run your solution, or if you want to document your solution, you have included a `README.md` file with the desired information.
- ☐ You have not shared your solution with anyone else or uploaded it to any publicly visible website like Github or Gitlab.
- ☐ You have included both the source and compiled code (if applicable) in a compressed bundle. If your code is in the folder `calendar_challenge` you can do this by running `zip -r calendar_challenge.zip calendar_challenge`.
- ☐ Most email providers don't let you attach code archives for security reasons. To get around this, please upload the zipped bundle to a location where it can be easily downloaded. We recommend CloudApp getcloudapp.com. Include the link to the download in your reply email to your recruiter.

We hope you enjoy this challenge! 🐦

Thought Machine - Front-End Engineering