

HTML5: Student's Choice
Extra Credit

due by noon ET on Thu 5/12

Ingredients.

- HTML5
- JavaScript
- ...

EXTRA CREDIT



Academic Honesty

All work that you do toward fulfillment of this course's expectations must be your own unless collaboration is explicitly allowed by some project. Viewing, requesting, or copying another individual's work or lifting material from a book, magazine, website, or other source—even in part—and presenting it as your own constitutes academic dishonesty, as does showing or giving your work, even in part, to another student.

Similarly is dual submission academic dishonesty: you may not submit the same or similar work to this course that you have submitted or will submit to another. Nor may you provide or make available your or other students' solutions to projects to individuals who take or may take this course (or CSCI.S-76) in the future.

You are welcome to discuss the course's material with others in order to better understand it. You may even discuss problem sets with classmates, but you may not share code. You may also turn to the Web for instruction beyond the course's lectures and sections, for references, and for solutions to technical difficulties, but not for outright solutions to problems on projects. However, failure to cite (as with comments) the origin of any code or technique that you do discover outside of the course's lectures and sections (even while respecting these constraints) and then integrate into your own work may be considered academic dishonesty.

If in doubt as to the appropriateness of some discussion or action, contact the staff.

All forms of academic dishonesty are dealt with harshly.

Grades.

Your work on this project will be evaluated along four primary axes.

Correctness. To what extent is your code consistent with our specifications and free of bugs?

Design. To what extent is your code written well (*i.e.*, clearly, efficiently, elegantly, and/or logically)?

Scope. To what extent does your code implement the features required by our specification?

Style. To what extent is your code readable (*i.e.*, commented and indented with variables aptly named)?

Student's Choice.

- Your task, by the project's deadline, is to implement the project whose proposal was approved by your TF! In addition to the features you have required of yourself, we do have some additional specifications.

Requirements.

- By now, you should've heard back from your TF either approving your project or requesting changes to the proposal. You should not proceed with your project's implementation until it has been approved. If it's not yet approved, do feel free to drop your TF a note to inquire its status. Reasonable changes to your app are allowed with approval from your TF.
- Along with your project, you will need to provide some documentation so we know how your project works and what it should do. Once your project is complete, write a document called `README.pdf` or `README.txt` that describes your app's usage, its purpose, and any setup that's required to get it working properly. In essence, reading through this documentation should provide us with enough information so that we understand your app and the entirety of its usage without needing to ask you any questions.

Implementation Details.

- Your app must render and behave properly in a Webkit-based browser.
- Your app's UI should be designed for a smartphone whose width is defined by `device-width`; its actual resolution might be anywhere from 320×480 to 760×1280.
- Your HTML must be valid HTML5, per <http://validator.w3.org/>.
- Your CSS and JavaScript must not be minified.
- You are welcome to integrate third-party JavaScript libraries into your project so long as you somehow cite their origin, per the syllabus.
- Under no circumstances should we be able to trigger runtime errors in your JavaScript code. Be sure that you handle unwanted inputs and HTTP failures elegantly, as by reporting such errors or silently handling. Under no circumstances should your code trigger errors in Webkit's own console.

How to Submit.

- First create a ZIP file containing all of your code (and any libraries you downloaded on which it depends) named #####.zip, where ##### is your 8-digit Harvard ID (HUID), the same credential that you use to log into `help.cs76.net`. Be sure that any paths in your code are relative, so that, when unzipped, we can access your app simply by opening `index.html` on any computer.

Then head to `https://www.cs76.net/submit`, click the **login** link at top-right, click the link to your TF's dropboxes at top-left, click this project's own folder, click **Upload File**, and upload your ZIP file as prompted; no need to give it a title. Be sure not to click the wrong project's folder. You may re-submit in this same manner as many times as you'd like. Just take care to delete any prior submissions.

Be sure not to submit or re-submit after this project's deadline.

EXTRA CREDIT