

Assessment: Section 1



[Download code](#)

Warning: Assessments

Remember, assessments are meant to be completed **by you**, not as a shared exercise with friends or other members of your cohort.

All code submitted should be **written by you**. If you incorporate code from elsewhere, it must be properly attributed.

Please do not put your assessment on GitHub.

It’s Jeopardy!

Using **jQuery** and **AJAX**, you’ll build a small, straightforward Jeopardy game.

Before you start, read about the [jQuery Service API](#), which provides categories and clues from the televised Jeopardy show.

Requirements

You can see a working version of this at <https://jeopardy-example.surge.sh/>.

- The game board should be 6 categories across, 5 question down, displayed in a table. Above this should be a header row with the name of each category.
- At the start of the game, you should randomly pick 6 categories from the jQuery Service API. For each category, you should randomly select 5 questions for that category.
- Initially, the board should show with **?** on each spot on the board (on the real TV show, it shows dollar amount, but we won’t implement this).
- When the user clicks on a clue **?**, it should replace that with the question text.
- When the user clicks on a visible question on the board, it should change to the answer (if they click on a visible answer, nothing should happen)
- When the user clicks the “Restart” button at the bottom of the page, it should load new categories and questions.

We’ve provided an HTML file and CSS for the application (you shouldn’t change the HTML file; if you want to tweak any CSS things, feel free to).

We’ve also provided a starter JS file with function definitions. Implement these functions to meet the required functionality.

Note: Randomly picking multiple things

In the requirements, we’ve asked for 6 random categories. Unfortunately, the jQuery Service API doesn’t have a method that returns a random category — you’ll need to figure this out.

There are a few possible strategies here:

- Get a bunch of categories, and keep randomly choosing one, making sure you don’t choose the same one twice.
- Get a bunch of categories, shuffle them, then pick the first 6. Unfortunately, Javascript doesn’t have a built-in shuffle function, but you can find hints online on how to make one.
- Find a function that will pick *n* random things for you. This is often called “sampling”. There’s a popular library for Javascript, Lodash, which provides a function that can sample a particular number of items from a larger list, making sure there are no duplicates.