3.5.2 Preview

To figure out how to implement the beta test suggestions, let's start by trying to answer the following questions:

- How do we handle a blank or null player name response?
- Are there other responses that may need a similar treatment?
- How can we handle mixed-case input to the fight-or-skip prompt?
- How do we update the `shop()` function to accept integer inputs?
- What method should we use to randomize the fight order?
- How do we save our high score in the browser storage?

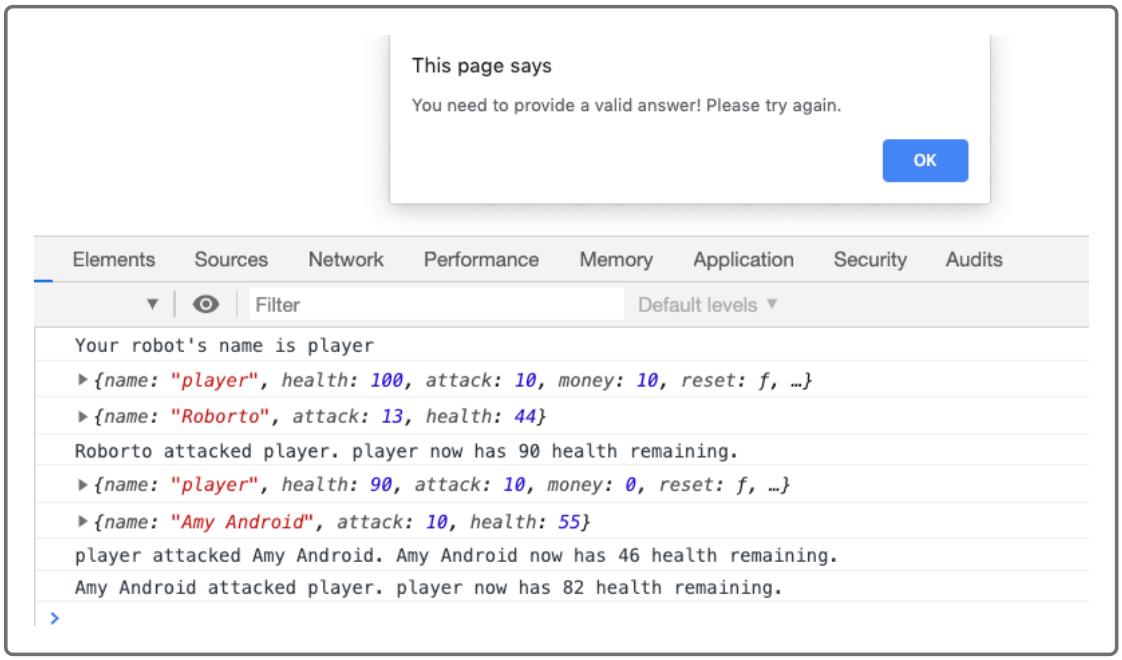
Here are the answers to these questions, which will guide the build process and help us create our GitHub issues:

- If a blank or null value is received for a player name, repeat `prompt()` until an acceptable answer is received.
- Extend the null input validation check to the fight-or-skip prompt as well.

- Make the response to the fight-or-skip prompt case-insensitive.
- Use conditional statements that execute based on the response's numeric value in the `shop()` function.
- Randomize the fight order for every round with a new enemy robot.
- Save our high score using the Web Storage API for `localStorage`.

SHOW HINT

Here's an example of the game's interaction in the Console panel of Chrome's DevTools:



The image above reveals an alternating pattern of attack between the player robot and the enemy robot between round 1 and round 2. Roboto attacks first in round 1 whereas the player robot attacks first in round 2. This variability demonstrates the fight order randomization.

Also observe that the alert message notifies the user, the answer provided is not valid. This message occurs when the user does not enter a valid numeric value when choosing options in the `shop()` function. The user is then navigated back to the options screen in the `shop()` prompt.

Here's a high-level overview of what we'll do in this lesson:

1. Create GitHub issues for every bug and feature request.
2. Handle blank, null, or mixed-case user input.
3. Fix the prompt bug in the `fight()` function.
4. Update the `shop()` function to handle integer inputs.
5. Randomize the fight order.
6. Save and load the high score from localStorage.

Are you ready to take Robot Gladiators to the next level? The clock is ticking, so let's get to it!