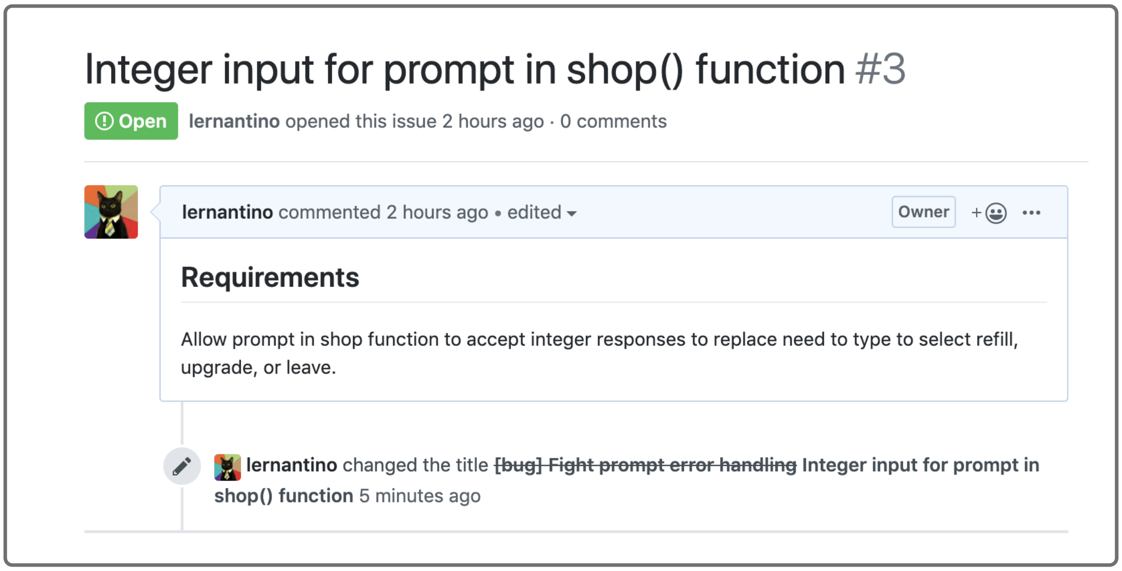
3.5.6 Update the shop() Function to Handle Integer Inputs

Let's look at the next issue:



Integer input for prompt in shop() function #3

lernantino opened this issue 2 hours ago · 0 comments

lernantino commented 2 hours ago • edited

Requirements

Allow prompt in shop function to accept integer responses to replace need to type to select refill, upgrade, or leave.

lernantino changed the title ~~[bug] Fight prompt error handling~~ Integer input for prompt in shop() function 5 minutes ago

Our beta testers reported that they'd like to do less typing for the prompt responses. Let's go ahead and create a branch for this called `feature/shop-input`. Next, let's think about how to proceed:

1) Change the conditional statements to execute code based on abbreviated commands.

2) Tell the user which abbreviated commands perform which actions.

Change the Switch Statements

Currently our `switch` statement in the `shop()` function checks our prompt response, `shopOptionPrompt`, for the commands to refill, upgrade, or leave. Let's replace these with numerical values so that a 1 will call the `refillHealth` method, 2 will call the `upgradeAttack` method, and 3 will leave the store. We'll keep our default case the same.

Let's change the case statements in `switch` to reflect these changes. On your own, replace the case statements as needed to only accept integers.

The conditional statements should now look like this:

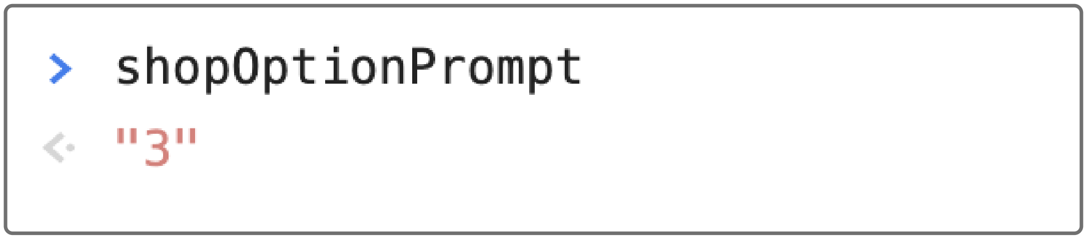
```
// use switch case to carry out action
  switch (shopOptionPrompt) {
    case 1:
      playerInfo.refillHealth();
      break;
    case 2:
      playerInfo.upgradeAttack();
      break;
    case 3:
      window.alert("Leaving the store.");
      break;
    default:
      window.alert("You did not pick a valid option. Try again.");
      shop();
      break;
  }
```

Let's save the `game.js` file and refresh the `index.html` file in the browser. Play the game and enter the Shop when prompted, and then try using numeric input instead of text input (e.g., "1" instead of "refill", etc.) It appears that our changes have created a different bug now so that neither the string command nor the integer is able to execute the statements we desire. Clearly the cases in our switch statement are not evaluating the responses as we expected.

The `switch` statement is no longer able to satisfy any of the cases and slips to the default case, re-prompting our request. Let's employ the DevTools debugger to take a closer look at the problem. Add the `debugger` keyword immediately before the `switch` statement:

```
// use switch case to carry out action
  debugger;
  switch (shopOptionPrompt) {
    case 1:
```

Reload `index.html` and play the game until you're able to enter the shop. Enter the shop and the DevTools debugger should pause execution. In the console panel, type `shopOptionPrompt` to see its value:



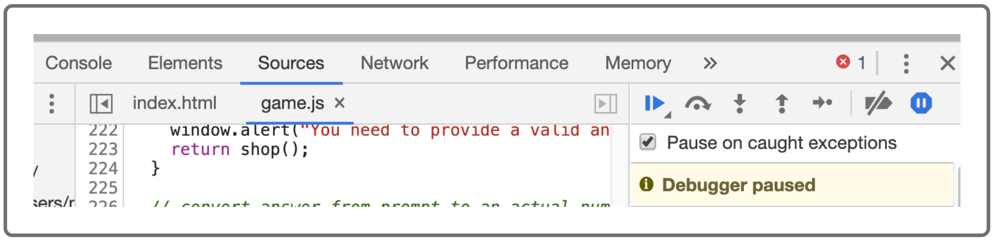
```
> shopOptionPrompt
<· "3"
```

Do you notice anything peculiar about our return value? The 3 is actually not an integer but the string "3". Prompt responses will always be a string, regardless of what was entered into the input field. In order to satisfy our case statements, we need to change our string into an integer by performing a data type conversion. We'll learn how to do that next.

In the meantime, remove the `debugger` keyword we added before the `switch` statement. But leave the browser paused where it is; we're going to experiment in the console a bit.

HIDE PRO TIP

The infinite loop we just created is now impossible to break! But there is a way to stop the code in the Sources panel in DevTools by pressing the pause button in the debugger controls:



```
    window.alert("You need to provide a valid an
    return shop();
  }

  // convert answer from prompt to an actual num
```

☑ Pause on caught exceptions

Debugger paused

To resume code execution, click the play button on the debugger control menu.

Data Type Conversion

To make our switch case statements work, we need another method to help us convert the data type.

PAUSE

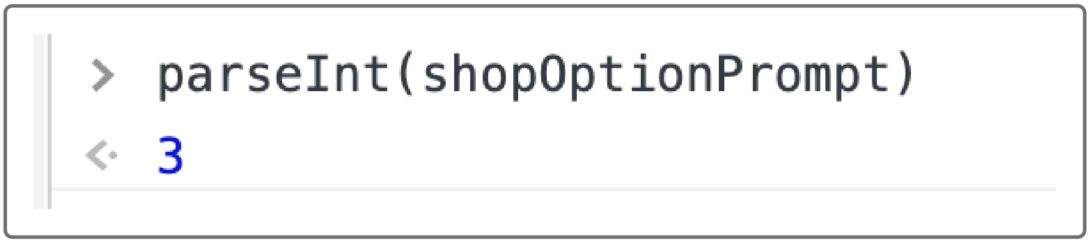
Search Google to find a built-in JavaScript function that converts strings to integers.

The `parseInt()` method converts strings to integers.

Hide Answer

Try it out by typing `parseInt(shopOptionPrompt);` in the console window of the browser.

The response should look like this:

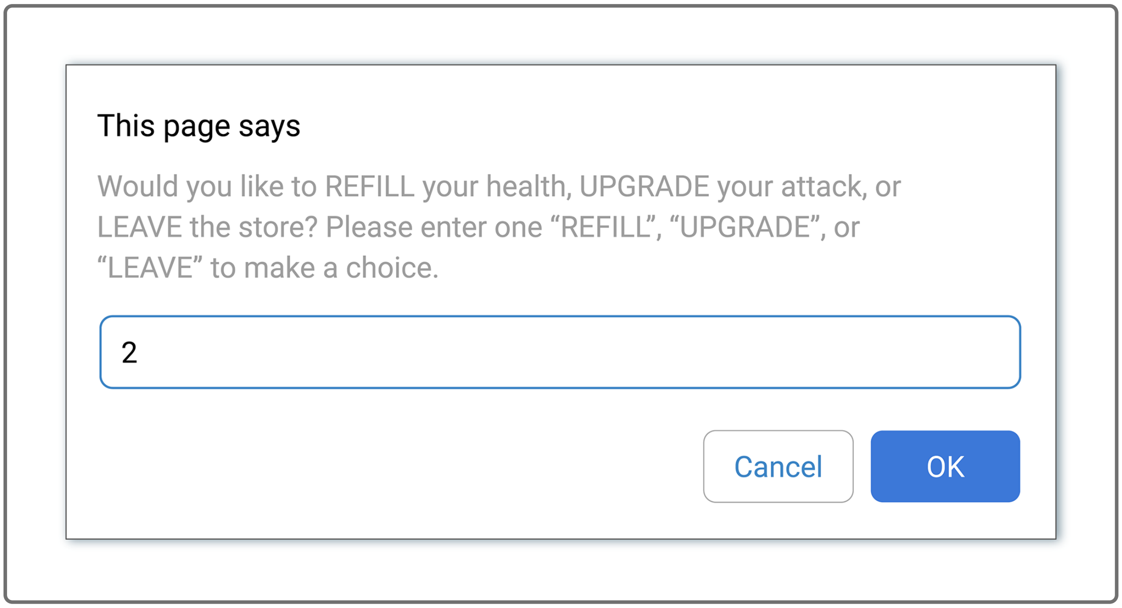


```
> parseInt(shopOptionPrompt)
<· 3
```

Let's add `parseInt()` to the `shop()` function before the `switch` case statement:

```
shopOptionPrompt = parseInt(shopOptionPrompt);
```

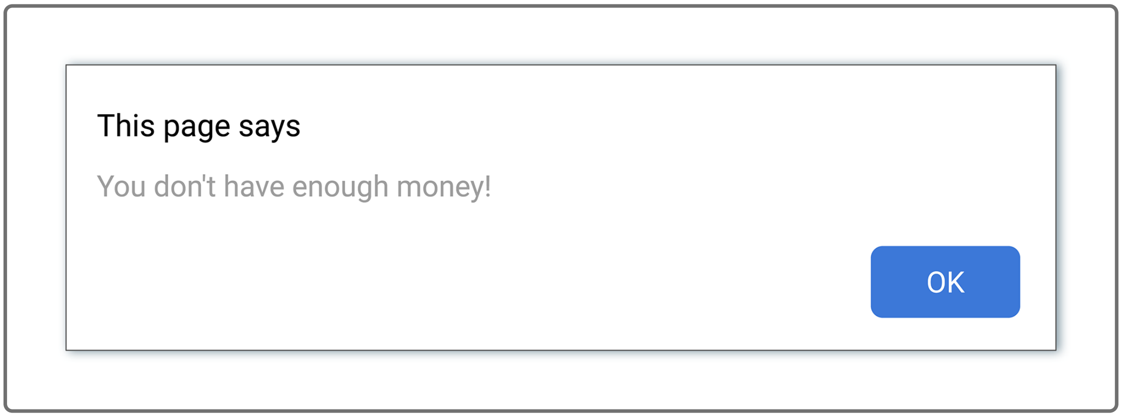
Save your work and refresh the browser to test the code. Once we get to the `shop()` function prompt, enter a numeric value to choose a shop option.



This page says

Would you like to REFILL your health, UPGRADE your attack, or LEAVE the store? Please enter one “REFILL”, “UPGRADE”, or “LEAVE” to make a choice.

2


This page says

You don't have enough money!

As the images above demonstrate, a numeric value was entered into the `shop()` prompt and was accepted as a valid input.

Nice job! Now the `shop()` function can accept numeric values. Change the prompt message verbiage to reflect the new data input.

```
var shopOptionPrompt = window.prompt(
    "Would you like to REFILL your health, UPGRADE your attack, or LEA
  );
```

Save and reload the `index.html` file in the browser. When you're satisfied with the result, add the feature to the `develop` branch and close this issue then move on to the next GitHub issue.