Your condition in JavaScript is incorrect because playerOneInput.value is always a string when retrieved from an input field, even if the user enters a number. The typeof check will always return 'string', so the condition will never evaluate to true.

**Correct Ways to Check If Input is a Number:**

1. **Convert the Value to a Number and Check If It's a Valid Number**
2. if (!isNaN(playerOneInput.value) && playerOneInput.value.trim() !== "") {
3. console.log("It's a number");
4. }
   * isNaN(value) checks if the value is **Not a Number**. We negate it (!) to confirm it's a number.
   * trim() ensures the input is not just spaces.
5. **Use Number() and typeof Properly**
6. if (typeof Number(playerOneInput.value) === "number" && !isNaN(Number(playerOneInput.value))) {
7. console.log("It's a number");
8. }
   * Number(playerOneInput.value) converts the string to a number.
   * typeof now correctly identifies it as "number".
   * We still need !isNaN() to check if the conversion is valid.
9. **Use parseFloat() for Decimal Numbers**
10. if (!isNaN(parseFloat(playerOneInput.value)) && isFinite(playerOneInput.value)) {
11. console.log("It's a number");
12. }
    * parseFloat() converts valid number strings into numbers.
    * isFinite() ensures it is not Infinity or NaN.

Would you like help improving your code further? 🚀



