Objective

Number Guesser

Overview

This project is slightly different than others you have encountered thus far on Codecademy. Instead of a step-by-step tutorial, this project contains a series of open-ended requirements which describe the project you'll be building. There are many possible ways to correctly fulfill all of these requirements, and you should expect to use the internet, Codecademy, and other resources when you encounter a problem that you cannot easily solve.

Project Goals

In this project, you'll write JavaScript functions to power a small guessing game. Your code will run in the browser instead of just the terminal, but you have an output section to help you test your functions and show you if you have syntax errors.

Setup Instructions

If you choose to do this project on your computer instead of Codecademy, you can download what you'll need by clicking the "Download" button below. You'll need to open and work in **script.js** in a text editor, and open **index.html** in a browser to test your code. If you need help setting up on your own computer, read our article about setting up a text editor for web development.

Download

Tasks 8/8 Complete

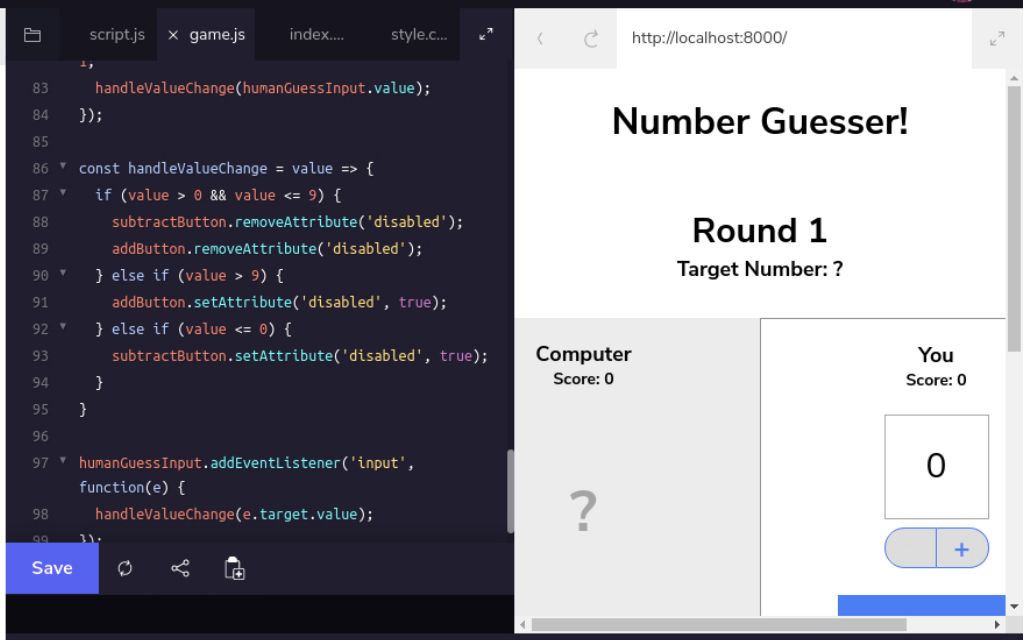
Mark the tasks as complete by checking them off

Prerequisites

1. In order to complete this project, you should have completed the first 3 sections of Introduction to JavaScript (through Learn JavaScript: Functions).

Project Requirements

2. In this project, you'll write four functions in **script.js**. We've provided some additional JavaScript code in **game.js** that will call your functions



based on user interactions, but you don't need to look at **game.js** and shouldn't edit it if you want your project to work as intended. As you complete this project, make sure that all of your functions are named exactly as specified so that they can be called correctly when the game is played.

This JavaScript project is probably different from most of the ones you've completed at Codecademy. Instead of just seeing text output from your program, your JavaScript functions are incorporated into a website that also uses HTML/CSS. You'll learn more about how to do this from scratch as you continue your JavaScript journey.

Experiment with a completed version of the project to get a sense of what you'll be building.

3. Create a `generateTarget()` function. This function will be called at the start of each new round in order to generate the new secret target number.

   This function should return a random integer between `0` and `9`.

Stuck? Get a hint

4. Create a `compareGuesses()` function. This function will be called each round to determine which guess is closest to the target number.

   This function:

   - Has three parameters representing the user (human) guess, a computer guess, and the secret target number to be guessed.
   - Determines which player (human or computer) wins based on which guess is closest to the target. If both players are tied, the human user should win.
   - Return `true` if the human player wins, and `false` if the computer player wins.

Stuck? Get a hint

5. Create an `updateScore()` function. This function will be used to correctly increase the winner's score after each round.

   This fu

This function:

- Has a single parameter. This parameter will be a string value representing the winner.
- Increases the score variable (`humanScore` or `computerScore`) by 1 depending on the winner passed in to `updateScore`. The string passed in will be either `'human'` or `'computer'`.
- Does not need to return any value.

6. Create an `advanceRound()` function. This function will be used to update the round number after each round.

   `advanceRound()` should increase the value of `currentRoundNumber` by 1.

   After completing `advanceRound()`, your Number Guesser game should be fully operational. You should be able to make guesses, see your or the computer score increase correctly, move to the next round, and see the correct round displayed.

Project Extensions & Solution

7. Great work! If you'd like to see the solution, move to the next task. If you'd like to extend your project on your own, you could consider the following:

   - You probably calculated the distance from the computer guess to the target and from the human guess to the target. Move this into a separate `getAbsoluteDistance()` function that takes two numbers and returns the distance, and then use that inside your `compareGuesses()` function.
   - Add functionality to check whether the user guess is between 0 and 9 and `alert()` the user that their number is out of range. It's not possible to set a number outside this range with the `+` and `=` buttons, but users can do so by typing directly in the input field.

8. Great work! Visit our forums to compare your project to our sample solution code. You can also learn how to host your own solution on GitHub so you can share it with other learners!

Your solution might look different from ours, and that's okay! There are multiple ways to solve these projects, and you'll learn more by seeing others' code.