Create a generateTarget() function. This function should return a random integer between 0 and 9.

The purpose of this function is to be called at the start of each new round in order to generate the new secret target number.

Stuck? Get a hint

**2.**

Create a compareGuesses() function. 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.

The purpose of this function is to be called each round to determine which guess is closest to the target number.

Stuck? Get a hint

**3.**

Create an updateScore() function. 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.

The purpose of this function is to be used to correctly increase the winner’s score after each round.

Stuck? Get a hint

**4.**

Create an advanceRound() function. This function should increase the value of currentRoundNumber by 1.

The purpose of this function is to be used to update the round number after each round.

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.

Stuck? Get a hint

**5.**

Test that your code is working properly by invoking your newly written functions within **script.js** with sample inputs. You can delete this code once you’re convinced that everything is working as it should.

Stuck? Get a hint

**6.**

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.

Stuck? Get a hint

**7.**

Great work! Visit [our forums](https://discuss.codecademy.com/t/number-guesser-challenge-project-javascript/462394) to compare your project to our sample solution code. You can also [learn how to host your own solution on GitHub](https://www.codecademy.com/learn/learn-git) 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.