The mandatory requirements for this app are as follows:

* You must use the HTML and CSS we’ve supplied. Once you’ve completed the project, you may choose to alter the layout and styles, but we ask that you stick with the templates we’ve supplied initially. Note that the index.html file already links to the CSS files, app.js file and jQuery. You should write your JavaScript code in app.js. Also, note that we already have inserted a small amount of code in app.js - there’s a $(document).ready() block with code that handles displaying and hiding the instructions for the game.
* When the page loads, JavaScript should start a new game. Since you’ll need to be able to start a new game when the user clicks the “New Game” button, you’ll want to create a newGame function that does everything necessary to start a new game.
* When a new game starts, a secret number between 1 and 100 should be generated that the user will have to guess. You should write a named function that takes care of this.
* The user should get feedback about each guess – if it was too low, too high, or just right. This means that you’ll need to write a named function that takes a user guess and determines which feedback to provide.
* Initially, you shouldn’t worry about telling users if they’re getting “hotter” or “colder” relative to their previous guess. Instead, you can use absolute values. For instance, you might decide that if a user is 50 or further away from the secret number, they are told they are “Ice cold”, if they are between 30 and 50 they are “cold”, if they are between 20 and 30 they are warm, between 10 and 20 hot, and between 1 and 10 “very hot”. We’ll leave it up to you to choose what the ranges are and what feedback you provide.
* Feedback about the guess should appear in div#feedback. By default, when the page loads, the text in this field is set to “Make Your Guess!”
* The game should track how many guess the user has made. Feedback about this should appear in span#count (which defaults to 0, when the page loads).
* The game should also supply users with a list of the numbers they have guessed so far. We’ve set up the CSS for this game in such a way that you can simply add each guessed number as an **<li>** to ul#guessList.
* You’ll need to ensure that users provide valid inputs. Note that the guess text input field has the [HTML 5 required flag](http://john.foliot.ca/required-inputs/) set, so you won’t have to worry about blank guesses being submitted (if the user submits a blank guess, they'll be prompted to supply an input). However, you will need to write code that ensures that the user has supplied a numeric input between 1 and 100.
* The starter template already contains a button in the upper right hand corner for starting a new game, however, this button does not currently do anything. You’ll need to write code that allows users to start a new game without making additional calls to the server. Clicking “New Game” should trigger the JavaScript function that starts a new game.

These are the minimal requirements. If you’re able to meet these basic requirements and want to build in more advanced features, you can write code that provides users with feedback about their most recent guess in relation to the previous one. If the most recent guess is closer to the secret number, you would tell the user they are “warmer”, and if it’s further, you’d tell them they are “colder”. Note that for the first guess, you’d still need to provide absolute feedback, since they’re won’t be a previous guess to compare to.

1. **Fork the Repository**: Make sure you’re signed into Github.com, then visit [our repo](https://github.com/thinkful-fewd/hot-or-cold-starter) and click on the “Fork” button in the top right. After your fork has been created, you’ll want to clone it to your local computer.
2. **Familiarize Yourself with The Layout and CSS**: You’ll need to use jQuery in the project to listen for when users submit guesses. You should spend a few minutes looking over the HTML and CSS files and using Developer tools to inspect page element.
3. **Break the App Logic Down Into Steps and Write Functions**: We already said that you’ll need a newGame() function that does everything necessary to start a new game. This function will itself need to call other functions to take care of specific tasks -- for instance, setting the randomly generated secret number. You should break the application logic down into discrete steps, then work on one step at a time.
4. **Share when You’re Done**: You should use Git/Github for version control on this project. Make sure to add and commit your changes as you work on the project, and once it’s complete, publish a copy using Github pages, and share a link to the live page and your repo with your mentor and fellow students.