

University of Arizona, CSc 337

Homework Assignment 3: Guess a Number Game

Special thanks to Marty Stepp of Stanford and Jessica Miller

This assignment tests your understanding of JavaScript and its interaction with HTML user interfaces. You must match the appearance and behavior of the following web page:

Guess a Number Game

Welcome to the CSC 337 guess a number game page! Enter a range into the boxes below and hit the start button to begin the game.

Minimum:
Maximum:

Your guess:

Your previous guesses:

This week you will create a page that allows the user to play a guess a number game with the computer. Your first task is to create a page [guessingGame.html](#) with a user interface (UI) for playing the game. No skeleton files are provided. Your page should link to a style sheet you'll write named [guessingGame.css](#) for styling the page. After creating your page, you must make the UI interactive by writing JavaScript code in [guessingGame.js](#) so that clicking the UI controls causes appropriate behavior. Your HTML page should link to your JS file in a `script` tag.

In total you will turn in the following files:

- [guessingGame.html](#), your web page
- [guessingGame.css](#), the style sheet for your web page
- [guessingGame.js](#), the JavaScript code for your web page

Turn in files separately on Gradescope. Do not submit them as a zip.

Our screenshots were taken on Windows in Chrome, which may differ from your system.

Appearance Details:

The page should have a title of **Guess a Number Game** sized 72pt with no margins. The preferred font for all text on the page is Helvetica or Arial in size 18pt. If that is not available it should use the default sans-serif font available on the system. The page should have no margins.

Under the page's heading is a paragraph with a width of 60% and margin on its left side of 10px. Next to this on the right side of the page is a div with a width of 20%. It has 30px of padding and a background color different from that of the rest of the page.

Below the paragraph and div are two divs placed side by side. The first contains the text, input and button for guessing and the other contains the previous guesses. Both have a margin on their left of 60 pixels. The text in the second that says, "Your previous guesses" should be in 20pt font, have no margin above it and be bold.

You may choose whatever colors you like.

Behavior Details:

When the page loads the minimum box should contain the number 1 and the maximum box should contain the number 10.

When the start button is pressed the program should choose a new random number between whatever is in the minimum box and whatever is in the maximum box. It should also get rid of any previous guesses or win information listed on the page

When the guess button is pressed the program should determine whether the number in the input next to the guess button is correct. If it is correct it should add “you got it right!” to the top of the list of previous guesses. Otherwise, if it was too low it should add “more than <number>” or if it was too high add “less than <number>”. Replace <number> with the number the user guessed. Guesses should be added to the top of the paragraph. You can add a
 tag between each hint to make them appear on their own lines.

Development Strategy and Hints:

1. Write the basic **HTML** content including the proper UI controls.
2. Write your **CSS** code to achieve the proper layout.
3. Write a small amount of **"starter" JS code** and make sure that it runs.
(For example, make it so that when the Start button is clicked, an **alert** box appears.)
4. Implement code to **generate a random number** when start is pressed.
5. Implement code to **read the users guess and print out a response** when guess is pressed

We strongly recommend that you install and use the **Firebug** add-on for Firefox on this assignment, or use the similar tool built into other browsers such as Chrome. Both show **syntax errors** in your JavaScript code. You can use either as a debugger, set breakpoints, type expressions on the Console, and watch variables' values. This is essential for serious JavaScript programming.

Our **JSLint** tool can help you find common JavaScript bugs. Since this is your first JavaScript program, you will probably encounter tricky bugs. If so, paste your code into JSLint to look for possible errors or warnings. For full credit, your JavaScript code must pass the provided **JSLint** tool with no errors reported. ("Warnings" are okay.)

For full credit, your **.js** file must be written in **JavaScript "strict" mode** by putting this exact line of code at the top:
"use strict";

Implementation and Grading:

Submit your assignment on Gradescope. All of your HTML, CSS, and JavaScript code should follow the **style guide** posted on the class web site.

Implement your page using **HTML5** as taught in class. Your page must pass the W3C HTML5 **validator**. Choose appropriate tags to match the structure of the page content. Do not express style information in the HTML page itself, such as inline styles or presentational HTML tags such as **b** or **font**. Use **unobtrusive JavaScript** so that no JavaScript code, **onclick** handlers, etc. are embedded into the HTML code.

Express all stylistic information on the page in **CSS** using your style sheet file. For full credit, your style sheet must successfully pass the W3C CSS **validator**.

Format your HTML, CSS, and JS to be readable, like to the examples in class. Place a comment header atop each HTML/CSS/JS file. Your JavaScript should have more descriptive **comments**, including a header on each function (including anonymous) and complex sections of code describing the relevant code, the function's behavior, etc.

You should follow reasonable style guidelines similar to those of a CSc 110 programming assignment. In particular, avoid redundant code, and use parameters and return values properly.

Minimize the use of **global variables**. Do not ever store DOM element objects, such as those returned by the **document.getElementById** function, as global variables. As a reference, our own solution has one global variable to remember the random number.

Format your code similarly to the examples from class. Properly use whitespace and indentation. In your HTML, do not place more than one block element on a line or begin a block element past the 100th character. In your JavaScript, properly space/indent your code, and do not write any lines of code longer than 100 characters.

You should not use any external **JavaScript frameworks or libraries** such as jQuery to solve this assignment.

Do not place a solution to this assignment on a public web site.