

PROG2001 – WEB DESIGN AND DEVELOPMENT

A-01 : *HI-LO* IN JAVASCRIPT AND HTML

OVERVIEW

You will write an application as part of an HTML file that will play a typical ***Hi-Lo*** game, where the player guesses values in a specific range until the correct number is found.

This is an individual assignment and must be completed on your own.

OBJECTIVES

This assignment supports the following course objectives:

- To practice writing HTML and JavaScript functions
- To use some of the events supported by the Document Object Model
- Become familiar with client-side validation and feedback techniques

ACADEMIC INTEGRITY AND LATE PENALTIES

- Please refer to the SET Policies document regarding [Academic Integrity Information](#)
- Please refer to the SET Policies document regarding [Late Policy](#)

EVALUATION

- Please refer to the assignment weighting in the *Instructional Plan* for the course as well as the assignment's Rubric in the course shell.

PREPARATION

Review Module-01 and Module-02 lesson content and also the Module-02 code samples – they will help you in this assignment.

REQUIREMENTS

1. The entire application must be written in one HTML page using only JavaScript and HTML.
2. Call this web page **a01.html**
3. Do not use any external CSS styling. This assignment is not about creating and using CSS. But if you do want to experiment with styles, define the style in the <HEAD> element or use in-line styles
4. When the page initially loads, the application should present a user prompt asking for the player's name. This prompt should be accompanied by a text box. [NOTE: Do not use the JavaScript prompt() function]
 - a. The player's name is mandatory and therefore cannot be blank – it can contain any non-blank characters.
 - b. An informative error message will be displayed for the user if they attempt to enter a blank name.
5. After the name is entered, the application shall hide the name prompt and will show a prompt to the user (using the user's name) to enter the *maximum guess* number [NOTE: Do not use the JavaScript prompt() function]
 - a. The application will only allow the user to enter an integer number and will ensure the value greater than 1. (the only real limit is the limit of an integer)
 - b. If any input other than this is given, then an informative error message must be presented
6. Your Hi-Lo application must create a *random integer* between 1 and the *maximum guess* number
7. At this point in the application, the *maximum guess* number prompt should be hidden as you enter the main runtime (*guessing*) UI of the application.
8. The following must be components of the main *runtime* UI:
 - a. A prompt asking the user (by name) to guess a number. This prompt must consist of a textbox to allow the user to enter a new guess at the random number. [NOTE: Do not use the JavaScript prompt() function]
 - b. A button to submit the guess labeled as **"Make this Guess"**
 - c. A message that informs the user about their allowable guessing range
 - i. For example, let's say the user enters 10 as their maximum number – your initial message would read "Your allowable guessing range is any value between 1 and 10."
 - ii. If the random number your code generated was 7 and the user enters a guess of 3, then after submitting their guess the message would change to "Your allowable guessing range is any value between 4 and 10"

- iii. This message must be present in the resultant HTML – do not use the JavaScript `alert()` function or any other pop-up messaging scheme
- d. If the user happens to enter a number outside of the allowable range – your logic needs to inform them of this and also simply restate your allowable guessing range message
 - i. For example, let's say the allowable guessing range is between 6 and 10 and the user enters the value 44 or -10 ... since they are number values, it's allowed – tell them it is outside of the allowable range and simply remind them of the guessing range
- e. During the game, if the user enters anything but a number, I expect your Hi-Lo game to remind the user that they need to enter a number – so display some kind of error message to that effect as well as the allowable guessing range message
- f. When the user's guess is correct, change the background colour of the page, hide the (main) guessing portion of the UI and show a message on the screen saying "You Win!! You guessed the number!!"
 - i. There should be no remnants of the main *guessing* prompt and/or messages
 - ii. At this point, show a "Play Again" button and reset your page to do it again
 - iii. Only this time you don't need to prompt the user for their name – but you should prompt them for a new *maximum number*.
- 9. Make sure that all data entered is validated for proper data type and in some cases proper data ranges/values
 - a. Remember that the objective of the code is to create a workable Hi-Lo game
 - b. Also remember that giving the user timely feedback is a mandated usability factor
- 10. The application should also only ever have one user prompt visible on the screen at a time
- 11. Make sure to comment your solution appropriately - HTML header comment, JavaScript function comment blocks as well as inline comments
- 12. Ensure that your Hi-Lo application runs properly and consistently within the Microsoft Edge and Chrome browsers.

FILE NAMING REQUIREMENTS

As mentioned above in the Requirements, the solution for this assignment must be developed and delivered in one (1) HTML page. Ensure that you name this HTML file “**a01.html**”.

SUBMISSION REQUIREMENTS

When submitting your single HTML page to the drop-box, **please ZIP it up first**. If you submit your HTML file (unzipped) to the drop-box, eConestoga may edit/adjust/alter your HTML source. Upload your ZIP'd solution to the appropriate drop-box by the deadline.