Jay Annadurai

MAD SP23

T. Faas

14 Feb 2023

Exercise 8: User Input

[Link to GitHub](https://jannadurai.com/Exercises/Exercise%208/index.html)

**Introduction**

     The HTML User Input exercises aim to increase student familiarity with user inputs and interaction with form elements. Specifically the exercise set targets the usage of inputs for user interaction, the retrieval of inputs via JS DOM methods, and then some function to modify or add value of the input.

**Algorithms & Planning**

     All files contain a similar structure of JS Classes followed by any general JS which is then ultimately followed by any JS functions. Within the specific exercise set, all exercises have the same program flow of defining a HTML object as a JS object, building a Console to store output of the program, and using signature functions to execute the manipulation of user input information into the desired output information.

**Reflection**

Ex8.1 is the basis of the other exercises. All exercises follow a similar structure of defining the HTMLasJS class that builds entities as JS representations of the HTML and then with methods, can create the HTML and manipulate it. An extended HTMLasJS class titled ‘Console’ builds a specific container element that inserts inner divs and contains methods to emulate a console that directly displays to the page rather than a log within the browser inspector. In the context of the User Input exercises, it served as the output for the final content. Ex8.1 itself is fairly simple—the logic involves fetching a value from a HTML input, executing the program logic of checking divisibility by seven, and then returning a Boolean in natural language. The program logic is also basic with the remainder operator as if the dividend input has a remainder when divided by 7, it is not perfectly divisible and thus will return a false which is then converted into a message. An interesting complimentary event keyup event listener is attached to the window such that if the number is divisible by seven, the input will turn green without an explicit user interaction.

Ex8.2 uses a very similar structure, but instead uses three user inputs- a lower bound input, an upper bound input, and a div that acts as a button to engage a random number generator that creates a number between the upper and lower bounds. The Console class is still used to log the final random number, as well as the upper and lower bounds for when the number was registered. The random number generation logic reuses the same logic from the midterm review exercises. It uses Math.ceil and Math.floor to adjust the random generation such that there are no rounding errors at the lower and upper bounds, and then does a linear adjustment to move the random generation interval to the correct interval.

     Ex8.3 is structured incredibly similarly to Ex8.1—a single input that has a single output. The only difference here is the program logic which instead removes all instances of the ‘#’ character from a string input. As such the input was changed from input of type number to a textarea. The function that removes the character ‘#’ simply uses the string prototype method ‘replaceAll’ and replaces the ‘#’ with ‘’ such that it emulates elimination of the character.

Overall, exercise set 8 was simple but did improve my confidence when working with user inputs. I chose the three most basic exercises to finish the exercise set quickly, however, as it’s the program logic for the I/O manipulation function that changes rather than the core concept, I’m confident I could still complete them. I also discovered I can directly implement HTML onevent attributes directly into my HTMLasJS class and I could also implement the program statements to limit the console log size directly into the class and then call the static method once within the event listener where I use the Console rather anything convoluted.