Hands-On #1

Adding Validation to MPG Application

Add the validations that are specified in this hands-on session, and test the application in your favourite browser. If the application doesn't run, find the error(s), fix them and re-run the application.

1. Using your text editor open the file “mpg.html”

2. Run the application with valid entries, and note the result. Then run it with invalid entries like zeros or spaces and record your observation

3. Add a selection statement that validates the miles entry after the statement that parses this entry (i.e. data). This statement should check if the entry is not a number or if the entry is less than zero. If either of this condition is true, display an appropriate message.

4. Add a selection statement that validates the gallons entry after the statement that parses this entry. Repeat some thing similar to the previous task.

5. Add a do...while loop for each entry that contains the statements that get, parse and validate the entry. This loop should continue until the entry is valid. To do that you can use the same condition in the while clause that you used in the if statement.

Creating a Sum of Numbers Application

This application prompts the user to enter a number between 1 to 100. Then the application adds all the numbers from 1 to the user's entered number and displays the sum of the numbers.

For example if the user enters 5, the application adds 1, 2, 3, 4 and 5 so it displays 15

If the application doesn't run, find the error(s), fix them and re-run the application.

6. Using your text editor open the file "sum.html"

7. Write the code for the prompt method that gets the user's entry. It should have a default value of 5 and a prompt message like "Enter top number to sum :". Then parse the entry into an integer value.

8. Declare a variable with a starting value of zero that will hold the sum of the numbers. Then code the for statement that calculates the sum of the numbers. This statement should use a counter variable with a starting value of 1.

9. Code an alert statement after the for statement that displays the calculated sum.

10. Add a selection statement that validates the user entry. This statement should check if the entry is not a number, if the number is less than 1, or if it is greater than 100. If any of these conditions is true it should display an appropriate message.

11. Add a do...while statement that will continue prompting the user for an entry until an entry is valid.

12. Improve the display of the result so the message looks like this:

Sum of numbers from 1 to 5 is 15