Part 1 - Setup
1. For this assignment you will have multiple files, so I would highly recommend creating a lab7 folder.
2. Inside of the lab7 folder you will run the following commands:

```
touch lab7.html script.js style.css
```
This will create all three files you will need for this lab with one command.

Part 2 - Find Multiples
lab7.html:
1. Make sure to source your script.js and style.css files.
2. In the body, you will have a container div with two main elements inside of it:
   ○ An <input> element with the id ‘userInput’
   ○ A <div> element with the id ‘multiples’
   ○ You may also want to include some headers to label each element.

script.js:
1. Use getElementById and addEventListener to add a “keyup” event to the ‘userInput’ element. This event will call the `generateMultiples` function you will define later.
2. Define a function called `generateMultiples()` that will do the following (in this order):
   a. IF `this.value` is blank OR `this.value` is not a number:
      • Update the `innerHTML` of the ‘multiples’ element with an error message
      • End the function early using a `return;` statement.
   b. Pass `this.value` to the parseInt function, and assign the returned value to a variable called `base`
   c. Create an array called `nums` with the values 1 through 12
   d. Use a for loop to iterate through the `nums` array.
      For each element in the array you will add the following to the `innerHTML` of the ‘multiples’ div:
      • multiply `base` by `nums[i]` to get each multiple value
      • use string concatenation to place each multiple value inside of a paragraph element
      • IF `i` is even, make the class for the `<p>` element ‘a’
      • IF `i` is odd, make the class for the `<p>` element ‘b’

style.css:
1. Set the width of the container div and center it on the page.
2. Add a text shadow to the headers/labels
3. Apply some interesting style/color to the id `userInput`.
4. Apply a different style to class `a` and class `b` so that every other multiple stands out. Even though they will have different styles, they should still be cohesive/complimentary.
Completed Example

Enter a number: 5

Multiples:
- 5
- 10
- 15
- 20
- 25
- 30
- 35
- 40
- 45
- 50
- 55
- 60

Enter a number: bad input

Multiples:
ENTER A VALID NUMBER!