Assignment: Homework 6 - Objects

- Due Tuesday by 11:59pm
- Points 15
- Submitting a text entry box or a website url
- Available Oct 29 at 5am - Dec 13 at 11:59pm

Homework 6 - Objects

What you will do:

- Create a main web page for Homework 6 with the following:
- a heading Homework 6 - Objects
- a hyperlink to Part 1 and a short summary of that part
- a hyperlink to Part 2 and a short summary of that part
- a hyperlink to go back to the main page of the course (Homework 1 web page)

DO NOT USE ALERT BOXES FOR ERROR MESSAGES OR RESULTS!!!

For Part 1:

- You will create a web page with
  - a heading that you can choose
  - a form that will have:
  - an input box for the user to enter a number with 4 decimals
  - a button to submit the data
  - a button to clear the form
  - a textarea (or another element outside the form - div or paragraph) where you will write the output/result
  - a hyperlink to go back to the main page of Homework 6
- You will create a script to resolve the following:
  - After the user types the number in the input box and clicks on the button to submit, you need to **make sure that the number has AT LEAST 4 decimal positions (no less than that)**
  - If the user types an invalid number, you will display an error message in the output/result area
  - If the user types a valid number (with 4 decimals and it is a number), you will use **Math.round, Math.sqrt, Math.floor**, **toFixed()** etc. to obtain the following:
    TIPS:
    when reading the input, make sure you will be using **parseFloat()** to not lose the decimal numbers before doing the calculations requested
    remember to use the **toFixed()** only at the time you will present the final result
    A. round the floating-point number to the nearest integer
    B. calculate the square root of the floating-point number and round it to an integer
    C. round the floating-point number to the nearest tenths position
    D. round the floating-point number to the nearest hundredths position
    E. round the floating-point number to the nearest thousandths position
  - You will then present in the output/result area the original number typed by the user and the results you found for each calculation requested above

For example:

- If the user types **4.32**, you should **NOT** accept it, and you should write an error message such as: "**You need to type a number with at least 4 decimals, please try again**" and then you should not present anything in the output/result area.
- If the user types **4.3264**, you should then do the calculations and the output/result area would have something as:
  "***You typed number 4.3264***
  Rounded to the nearest integer = 4
  Square root rounded to integer = 2
  Rounded to the nearest 10th position = 4.3
  Rounded to the nearest 100th position = 4.33
  Rounded to the nearest 1000th position = 4.326"

For Part 2:

- You will create a web page with:
  - a heading that you can choose
  - a form with:
  - a textarea to allow the user to type a long content
  - an input box to allow the user to type only one letter
    TIP: in HTML, you can restrict the number of characters the user will type by using the attribute **maxlength** in the input type="text" element
  - an input box or textarea, or an element outside the form (a div or paragraph) to present the output/result
  - a hyperlink to return to the main page of Homework 6
- You will write a script **using some Script methods you learned**, to solve the following:
  - The user will type whatever content he/she wants in the textarea for the long content, and the user will also type a single character in the input box you provided. When the user clicks the submit button, you will:
    - Search for the character the user typed inside the content in the textarea and you will count how many times this character is shown in the big content
    - If the character **is found**, you will present a message in the output/result area that says **how many times the character X shows up in the content** where **X** is the single character typed by the user
    - If the character **is not found**, you will present a **new window** (using the **window.open()** method you learned), the message: "***Search character X not found in the content you typed***", where **X** is the single character typed by the user. **You should also make sure that, in this case, you will clear the input box or textarea that would be presenting the result of a positive search**
      NOTE: The new window should be **300 pixels** by **100 pixels** (width by height) and should be presented on a place that do not obscure any of the main content on the page the user is working with - you will choose the best place depending on your layout
      TIP: when presenting the new window, it will be GREAT, if you present a button that the user can click to **CLOSE** that new window

IMPORTANT NOTES:

Remember that **it does not matter** if the user types a lowercase letter to search - the match should be done and accepted if the text contains the same letter either in lowercase or uppercase. For example, **letter "t" would be found twice in this text "This time"!**

For a tip on how to use the **charAt()** to count the number of occurrences of a character, take a look at this page coded by the instructor.

To submit your homework:

1. Create the main page for Homework 6
2. Create one page for each part of the Homework 6
3. Update your Homework 1 main page of the course) to include a hyperlink to go to the Homework 6 page
4. Upload all the created and updated pages to the web server
5. Test the URL of Homework 1 page in the browser to make sure that the link to Homework 6 page is working and test the links to the parts of the assignment that are provided on the Homework 6 page
6. If everything is ok, you will submit the URL of your Homework 1 page and from that page I should be able to click on the Homework 6 hyperlink to start evaluating your work

Checklist for grading:

1. All your pages open in the browser with no errors of NOT FOUND or FORBIDDEN, you submitted the URL to Homework 1 page with the updated link to this assignment - **1 point**
2. The script of all the pages are working according to what was requested and with the validations requested - **12 points** (6 points for each)
3. The user interface/user experience is appropriate and error messages are not showing when the right output is being shown and/or the output is not being shown when there is an error in the expected value to be typed by the user - **1 point**
4. The HTML and CSS of all your pages are valid - **1 point**