Assignment 4 – HTML Form Validation with JavaScript
CGS 3066 - Spring 2019

Deadline: Monday April 8, 2019, 11:59 PM.

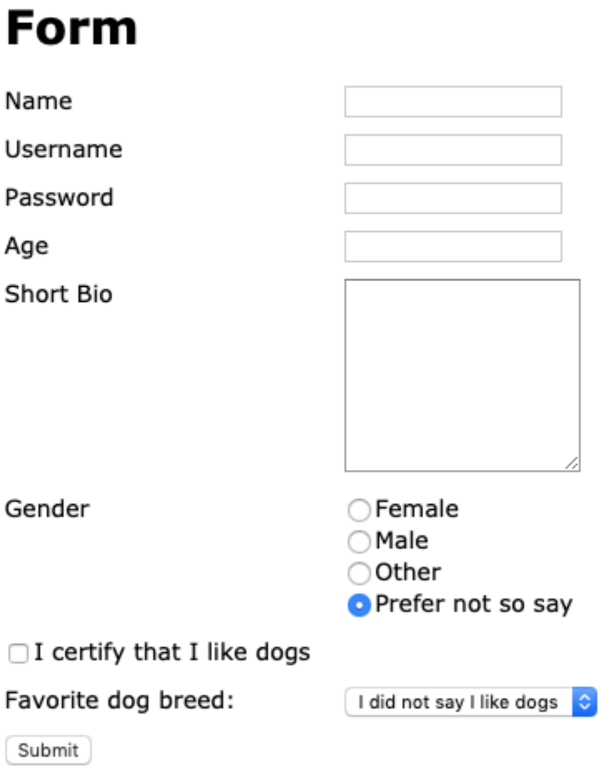
Objective: Familiarize with HTML form validation using JavaScript

In this assignment, you will create validation rules for an existing HTML form. If the data entered by the user does not satisfy any of the validation rules, the web page will bring appropriate error messages.

Instructions:

1. Download the a4_template.zip file. Extract the compressed file to your machine.

2. The template will contain an index.html file, a CSS file, and an empty JavaScript file. The following is a screenshot of the web site as soon as it is opened in the browser:



Form

Result

Form not submitted yet

Name

Username

Password

Age

Short Bio

Gender
○ Female
○ Male
○ Other
● Prefer not so say

□ I certify that I like dogs

Favorite dog breed: I did not say I like dogs

Submit

On the left side, the web page displays a form that the user will complete and submit using the Submit button. After the form is submitted, the web page will react in one of the two following ways:

a) The Result column (the column of the right side) will display the information entered by the user. This is considered a successful form submission
b) The web site will inform the user about all the validation rules the data entered has violated. In this case, the Result column will not display anything. This is considered a failed form submission.

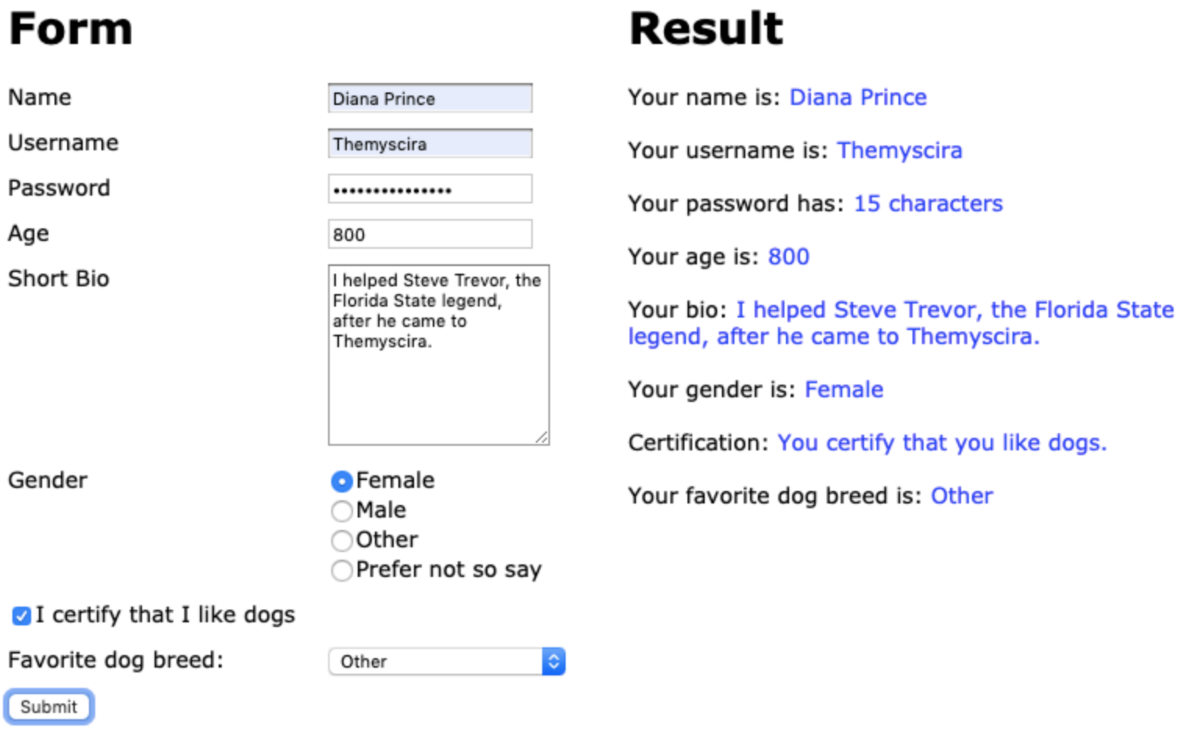
3. Implement, using JavaScript code, the following validation rules for the form:

   Points: 70

   a. All the fields that receive text must be completed (i.e., Name, Username, Password, Age, Short Bio) are required **(10 points)**.
   b. Name must have its first letter in uppercase **(10 points)**.
   c. Username must have at least 5 characters **(10 points)**.
   d. Password must have at least 6 characters **(10 points)**.
   e. Age must be a number greater than 0 **(10 points)**.
   f. Bio must contain the strings “fsu” or “florida state”. This rule is not case sensitive **(10 points)**.
   g. If the user does not check the *I certify that I like dogs* checkbox, then the user is not allowed to select a favorite dog breed (i.e., in that case, the user can only select the *I did not say I like dogs* option) **(10 points)**.

4. In case the data entered by the user does not satisfy any of the previous validation rules, **the web page must inform/alert the user with the specific rule that was not satisfied.** That is to say, generic error messages are not allowed. You can implement this feedback for the user using any strategy **(10 points)**.

5. If the data entered by the user passed all the validation rules, the web page must display the entered information using the *Results* column. The following is an example of the expected result column after the user makes a successful submission: **(10 points)**.



Form

Name	Diana Prince
Username	Themyscira
Password	•••••••••••••••
Age	800
Short Bio	I helped Steve Trevor, the Florida State legend, after he came to Themyscira.
Gender	● Female ○ Male ○ Other ○ Prefer not so say

☑ I certify that I like dogs

Favorite dog breed: Other

Result

Your name is: Diana Prince

Your username is: Themyscira

Your password has: 15 characters

Your age is: 800

Your bio: I helped Steve Trevor, the Florida State legend, after he came to Themyscira.

Your gender is: Female

Certification: You certify that you like dogs.

Your favorite dog breed is: Other

6. Use good programming practices **(10 points)**.
    a. Your name at the beginning of each file you submit
    b. Good indentation
    c. Meaningful variable and method names
    d. Code comments

Total points for this assignment: 100 points

Extra Credit:

- Fix the bug in the code that lets the user to select "Male" and other gender at the same time **(5 points)**.
- Give feedback to the user when they do not enter valid data. In particular, use an error message located **next to the field that contains the invalid data** (it can be located on the right side or right below the field). For instance, if the user enters a username that has less than 5 characters, the error message must be shown below/next to the username input element. Use this strategy for all the validation rules you need to include for this assignment **(10 points).**
- Instead of triggering the validation rules when the user submits the form (i.e., when they click the *Submit* button), trigger the validation rule as soon as the user enters the data. Therefore, as long as a field has invalid data, a specific error message should be displayed for the field. Then such error message needs to disappear as soon as the user finishes entering the right data for the field. For instance, assume that the user wants to enter "Themyscira" as their username. When they type "T" or "Th", the web page must immediately inform them with an error message saying that the username has less than 5 characters. Such error message must disappear as soon as they enter "Themy" in the username field. You are not allowed to use alert() for this extra credit item **(15 points)**.

Submission format:

Submit to Canvas a compressed file containing the modified version of the template. Make sure your submission includes all the code you wrote for this assignment.

Late submission policy:

As described in the syllabus, any late submission will be penalized with 10% off after each 24 hours late. After 5 days late the grade is 0 (zero).