Siobhan Lynch
Week 5 Homework Questions Part 2:

1. What ***IS*** form validation?
   Form validation occurs on the servers end, after someone enters all required information and data in the correct format and hits the submit button. Then that person gets feedback about the info they entered in the form inputs.

2. What are the two types of client-side form validation?
   Two types of client-side form validation are html5 built in form validation and the constraint validation API.

3. What exactly is html5 built-in form validation? Describe the one ***attribute*** that needs to be **added** to the ***form input element*** in order for **built in form validation** to ***work*** at a **minimum**?
   Html5 built-in form validation can validate most user information without using Javascript. An attribute that needs to be added to the form input element is the input boolean attribute.

4. Use the different input elements (username, password, email) we have talked about and demonstrated in class as examples of how to implement html5 built-in form validation. You can and SHOULD take from the form which you have to build for your next project as examples. That's the idea here. A number of you used code from the Arithmetic Forms to provide examples in answers to the questions for the week 3 homework. That was really excellent! I would like to see the same happen here.

```
<input type="text" name="username" placeholder="create
username" pattern="^(?=.*\d)(?=.*[a-zA-Z]).{7,15}$"
required>
<input type="email" name="mail" placeholder="enter your
email"
pattern="(\b[a-z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}\b)"
required>
<input type="password" name="password"
placeholder="create a password"
pattern="^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])\w{13,24}$"
required>
```

5. What is the main difference between HTML5 form validation and the JavaScript Constraint Validation API?
   The difference is that HTML5 form validation doesn't use JavaScript while constraint validation API does.

6. What is the input attribute you would NOT want to use when creating a form with the JavaScript Constraint Validation API?
   The input attribute that we would not use is the "title attribute" because it overrides any messages set up via JavaScript using the ".setCustomValidity( )" method.

7. Is client-side form validation enough by itself? Why ***not***?
   Client-form validation isn't enough by itself because then the inputs are not secure. and Client side validation is easily hacked. That's what website security is for, to prevent users' information

from being leaked and exposed. Security checks can only be done server-side so there is an extra necessary step.

8. Why would we want to use form validation when at the same time we want them to have the best UX possible on our site? Give ***three reasons*** why we would want to use **form validation** on our users.

   We want to use form validation and have the best UX design on our website for 3 reasons. One, we want all the client's information in the correct format. Second, we want to protect the users' information by forcing the users to create a secure password to protect the information they input. And third, we want to protect our site from hackers.

9. What is one of the most significant features of HTML5 form controls? How is this achieved?

   The most significant features of HTML5 form controls is that it can validate most user information without using Javascript. It's achieved by using validation attributes on form elements.

10. List **seven attributes** used in **built-in form validation**, and ***describe*** what they **do**.

    required: Specifies whether a form field needs to be filled in before the form can be submitted.
    minlength and maxlength: Specifies the minimum and maximum length of textual data (strings)
    min and max: Specifies the minimum and maximum values of numerical input types
    type: Specifies whether the data needs to be a number, an email address, or some other specific preset type.
    pattern: Specifies a regular expression that defines a pattern the entered data needs to follow.