Reading Notes--- Week 10

Client-Side Form Validation—allows you to check input data before it is sent to the server. Doing this reduces error and improves the user experience by speeding up the processes.

What is Form Validation?
Give the user feedback when they enter something unexpected…such as not an email address or a phone number when one is required.

Validation is done in the browser---called client-side validation.

--It will ensure you are getting the right data.
--it will protect the user and account info.
--It protects the webpage form malicious users.

Different types of client-side Validation
--built in validation…done through the HTML…doesn't require much JS
--JS validation is done using JavaScript.

Built-in Form Validation
--Done by using attributes on forms through HTML.
Such as: required, minlength, min, pattern, etc.
CSS pseudo-classes include :valid, :invalid, :out-of-range, :required, etc.

Validating Against regular Expressions
Pattern attribute—uses regex(regular expression)
--a = a and mothering else such as "aa" or "b"
--abc = "abc" only
--ab?c = a and then b or c…"?" = or
--ab*c means that "b" could have 0 or multiple "b"s…such as bbbbb or no "b"s
--a|b---means a or b.

Constraining the length of your entries
--minlength and maxlength---good for <textarea> or <input>

Constraining the values of entries
--min or max---use for input "type = 'number'" only.

VALIDATING FORMS USING JAVASCRIPT

Constraint Validation API—supported by most browsers. Has properties and methods for each of the HTML elements such as "button, fieldset, input, select, etc.

Properties available include :
validateMessage, validity (with several properties), and willValidate.

The Constraint Validation APR also makes the following methods available: checkVAlidity, reportVAlidity, preventDefault, and setCustomVAlidity

--validity properties:
patternMismatch, tooLong, maxlength, tooShort, minlength, typeMismatch, etc.

Implementing a customized error message:
You can implement a customized error message which can't me done with HTML attriabutes.
You can also change the look and feel of the message with CSS by using the JS method.

Use "novalidate" attribute in form to disable the HTML validation.

How to decide the best method for Validation? Ask these questions:

What kind of validation should I perform?
What should I do if the form doesn't validate?
How can I help the user correct invalid data?

USING FETCH

Provides a JS interface for accessing and manipulating parts of HTTP pipeline…such as requests and responses.
Global fetch() method is a way to fetch asynchronously across the network.

Different from jQuery.ajax() in the following ways:
--the fetch method doesn't reject HTTP error status…only rejects if there is network failure to prevent the request from completing

--won't sent cross-origin cookies…unles you set the credentials "init" option.

Simpliest way is to take one argument which is the path you want to fetch.

Response object does not directly contain the JSON response body but represents the entire HTTP response.
--the json() method returns and resolves the results with parsing the body text as JSON

Supplying Request Options
-fetch() can optionally accept a second parameter…"init"

SENDING A REQUEST WITH CREDENTIALS INCLUDED

To send request with both same-origin and cross-origin is called credentials…use “include” to pass to fetch.

```
fetch('https://example.com', {
  credentials: 'include'
});
```

You can also use “omit” and “same-origin” for credentials.

UPLOADING JSON DATA
--the HTTP post method sends data to the server. The type of the body of the request is indicated by the Content-Type header…sent via HTML form.

UPLOADING A FILE— Files can be uploaded using an HTML <input type="file" /> input element, FormData() and fetch().

The FormData interface provides a way to easily construct a set of key/value pairs representing form fields and their values, which can then be easily sent—the FormData() constructor creates a new FormData Object.

You need to handle processing a text file line by line.

CHECKING THAT THE FETCH WAS SUCCESSFUL

Items to research and understand better:
--CORS errors
--async-await
--Class
--Constructor—what to put in it.
--ServiceWorkers
---catch vs .catch---when to use one or the other.
--blob
--POST method?