Web Front-End Development
Week 3: HTML Forms

HTML Forms

The most common way to get user input on the web is through HTML forms.
Although forms often submit data to a server, we can also use the input client-side.
We're going to look at how we can use JavaScript to access form data and create interactive web pages.

Forms group together controls

- All controls must be within **<form> </form>** tags
- Each form element is stored as an object
- The input object can generate several elements
  - **type** specifies the type of element
  - **name** allows you to name the element
  - **id** allows you to give each element a unique id

Basic Form controls

Text field and text areas
Text fields handle one line of text.
Text areas are similar to a text box but can contain multiple lines

Check Boxes
Check boxes allow a user to choose from a group of options

- Each check box is independent
- Users can check as many check boxes as they want

Radio Buttons
Radio buttons allow a user to pick one choice from a group of options.

- Each button has an id that must be unique
- Radio buttons are grouped by the name attribute so only one can be chosen at a time

Select/List Boxes
A select, or list, box lets the user chose from a list of options.
Good choice if there are a lot of choices(state)

Buttons
Buttons usually trigger an action when clicked.
Use an event listener to listen for the click event to call a function to process the form.

Labels
Buttons have text in the tag
For all other form elements use **<label>**

- The **for** attribute should reference the id attribute of its related element to bind them

Example:

- Pac-12 season tickets (form_select.html)
  - text field

- **input** is a self-closing tag
- **id** specifies the id of the text field
- **value** specifies the text that the text box initially contains (optional)
- **size** determines the size of the text box in number of characters (optional)

  - checkboxes
    - A label for the check box needs to be done separately in HTML
    - Property **"checked"** is **true** or **false**
  - radio
    - Radio buttons are grouped by **name** so only one can be chosen at a time
    - Each **id** must be unique
    - The value of the button is stored in **value**
    - **checked** stores if the button is checked
      - **true** for the chosen button
      - **false** for the ones not chosen
  - select
    - The **select** object creates a drop down list box
    - The **select** object is a container for a series of **option** objects
    - Each **option** object has a value property that stores the value of the selected option.
  - button
    - **type** describes what type of button it is
      - submit will submit the form to a server
      - button doesn't submit the form, by itself it does nothing
        - we'll use the event listener to handle the event
      - reset resets the fields in the form
    - **id** is a unique id
  - text area
    - The **rows** and **cols** values determine the size of the text area (in characters)(optional)
    - These objects have a value property that you can read and write to
- js/pac12e.js processes the form
  - add an event listener for the click event on the button
  - when the click event fires the pac12school function is called

HTML5 input types

Text fields were being used for so many different types of data so HTML5 added more input types
On some devices these input types will show different keyboards for the users (iOS)

<u>New input types</u>

- search
- email
  - requires well formatted email address (requires button type submit)
- url
  - requires well formatted url (including http://) (requires button type submit)

- tel
- number and range
  - min, max, step, value

    - The valueAsNumber DOM property returns the value as an int instead of a string which is useful for numbers and dates
- date, month, week, time
    - min, max
- color
    - will show either a default color picker or another widget depending on the browser.

<u>New attributes</u>

- required
    - Boolean attribute that specifies that an input field must be filled out before the form is submitted
    - works with the input types: text, search, url, tel, email, password, date pickers, number, checkbox, radio, and file
- autofocus
    - sets the focus on the element when the page is loaded (button must be type submit)
- placeholder
    - string for instructions
- autocomplete
    - on or off
- spellcheck
    - true or false

CSS has new pseudo-classes for the UI element states to help you style form elements based on their state.

You can make the form more user friendly by reducing the amount of input when possible.

- Disable or hide fields not needed

<u>Example:</u>
form_html5.html, pac12_html5.js

- Would you like to purchase season tickets?
    - yes shows the order form
    - no hides the form
- Name field is required
- Email type=”email” to check for valid email address
- Changed the button type to submit and replaced the click listener with a submit listener on the form
    - it checks the HTML5 validation before submitting
    - event.preventDefault() prevents page from being reloaded which is the default action
- Disables card number and expiration if using venmo
- Hides shipping info if the same as billing