Announcements

- No posting of code in the forum
- Check class announcements daily
HTML Forms

- Regarding the name property
  - The name property is used by servers to process the data
  - Most of the time we will set the name and id attributes to the same value
- Forms ➔ Previous examples of forms just collected data for processing by a JavaScript program
- Forms ➔ Means by which information passes from the user to a server
- Submit button
  - Sends form data to a server application.
  - It is this special functionality that other buttons don’t have
- <form tag> ➔ Two important attributes:
  - **action** ➔ indicates where the form contents will be sent
  - **method** ➔ defines how the contents will be sent
HTML Forms

- Method → Two alternatives
  - post
    - Contents sent using the HTTP POST method
    - Content is sent in the body of the message
  - get
    - Contents sent using the HTTP GET method
    - Contents included in the URL

**Example:** formGet.html, getProcessing.php

**Example:** formPost.html, postProcessing.php

- Important: You can only see the results if you have a web server running the .php files.
- Could you install a web server in your computer? Yes!
- **Windows →**
- **Mac →**
GET vs. POST

- **Get**
  - Information visible as it is displayed in the URL
  - Limited in the amount of information it can send
  - URLs associated with get can be bookmarked
  - Intended for operations that will not change the server state

- **Post**
  - Information is not presented in the URL
  - No limits on amount of information it can send
  - Because variables are not displayed in the URL it is not possible to bookmark the page
  - Intended for operations that will change the server state
  - Try reloading a form submitted using post. Why the warning?
Form Processing

- As we have seen, we need a server script/program to process the data we collect in a form and that is sent via submit.
- If you know the script/program to use you can develop your own customized form to provide a service (e.g., Google search).
- Make sure you name your form elements after the variables required by the script/program.
- Notice we are not using any JavaScript at the moment. We are focusing on what we can do with forms by relying on the server script/program for processing.
- **Example:** SearchBox.html
Form Validation

- Forms → We can validate the data associated with a form by recognizing the submit event.
- Keep in mind that JavaScript can be disabled therefore the server application must also validate the data.
- **Example:** formValidation.html, FormValidation.php
CheckBoxes

- Allow us to make a selection
- Defined by using type="checkbox"
- We can tell whether an entry is selected by using the “checked” property
  - true → entry has been selected
- Default selection by using checked="checked"

**Example:** Checkboxes.html
Radio Buttons

- Exist in groups and only one can be checked in a group
- Defined by using type="radio"
- We defined the radio buttons to be in the same group by using the same name for all of them
- Note: Do not use the same value for name and id
- Default selection by using checked="checked"
- We access the elements using arrays
- `document.getElementsByName` ➔ returns array

**Example:** RadioButtons.html
Drop-DOWN/Scrollable Lists

- Defined using the `<select>` tag (not the `<input>` tag)
- `<option>` tag to specify possible options
- To define a default choice use:
  ```html
  selected="selected"
  ```
- The `multiple` attribute in `<select>` allows for selection of multiple items (usually displayed as a multiple-selection list)
- **Example:** DropDownList.html
- **Example:** ScrollableList.html
Textareas

- Allows user to input more than one line of information
- We use the `<textarea>` tag
- `rows` and `cols` attributes define the text area
- Default text (if any) appears between the `<textarea>` and `</textarea>` tags
- **Example:** Textarea.html