Forms in HTML5

What are the form basics?

- Forms are used to pass data to a server
- Different input elements can be used
- Rely on **action** and **method** attributes to identify where to send the form data and how to process it.
- Forms must have server with them.

Form Tags

- <form>
- <fieldset>
- <legend>
- <label>
- <input>

```
<form action="#" method = "post">
        <fieldset>
                <legend>Personal Information:</legend>
                <label for="name">Name</label>
                <input type="text" id = "name"><br/>
                <label> Email:
                        <input type="email" value = "email@umich.edu">
                </label><br/>
                <label for="birth">Date of birth:</label>
                <input type="date" id = "birth"><br>
                <input type = "submit" value = "Save!">
        </fieldset>
</form>
```

Input Types

- Text Fields: one-line input for text
- Password: characters masked
- Radio Buttons: select ONLY ONE from a set of choices
- Checkboxes: select ZERO or MORE from a set of choices
- Textarea: multi-line input control
- Email:
- Drop-down list (with or without pre-selected value)
- Select:
  - <optgroup>
  - <option>
- Submit Button: used to send form data to a server
- Hidden fields
- Upload

Form elements have attributes

- **type**
- **name**
  - Used to pass value to the server
- **id**
  - Used to connect labels and/or JavaScript
- **value** attribute does different things depending upon the type.
  - *Button*: text inside the button
  - *Text field*: populates the field with default value
- **class**
  - Used for styling

STOP: Make a form

- Use the code from Canvas to make a form.

How can we augment this form?

- Better tags, better attributes, and better styling!!

- Include JavaScript!! (Coming soon….)

Back to the Request/Response Cycle

- Two parts:
  - Front end
    - What you are developing... What happens on the browser/client side
    - HTML, CSS, JavaScript
  - Back end
    - What the server is handling
    - Python, Ruby, PHP, Perl, Java

Back to the Request/Response Cycle

- Two parts:
  - Front end
    - What you are developing... What happens on the browser/client side
    - HTML, CSS, JavaScript
  - Back end
    - What the server is handling
    - Python, Ruby, PHP, Perl, Java
  - **Don’t forget the USER**
    - **User experience designer**

HTTP Methods

- **GET**
- **POST**
- HEAD
- PUT
- DELETE
- OPTIONS
- TRACE

http://www-personal.umich.edu/~collemc/form2.html

http://www-personal.umich.edu/~collemc/form3.html

Pressing Submit

- When you fill in a form and press “Submit” the browser packs up the parameters of the form and sends them to the server using the “name=” values as the parameter names and the field contents as the values.

- GET → Parameters are placed in the URL
- POST → The URL is retrieved and parameters are appended to the request in the the HTTP connection

Get vs Post

- GET is used when you are reading or searching for things
- POST is used when data is being created or modified

- Web spiders will follow GET not POST

- Can you give me an example of when you would want to use each?

Form Processing

- Unless you use Jekyll, we can’t process forms that are hosted on GitHub pages.

- Instead, I recommend formspree
  - https://formspree.io/

- When you use plugins like FormSpree you will see differences in local vs hosted sites.

More Input Types

- **number** (min, max, step)
- **tel** is for telephone numbers (pattern)
  - No restrictions, enforce a style with
- **url**
- **range**
  - No need for validation
- **search**
  - Typically just a textfield
  - Can be used with datalist for common terms
- **color**: color selector or regular text input
- **date**
  - When in auto-focus generates date picker
  - Fallback is regular text input

required

- Marks any input/textarea as being required to have a value before the form can be submitted.
- Any field using the required attribute must also have a name attribute.
- The first required field which has no value when a form is submitted will be forced into focus upon form submission (like a "hint" to fill it out).
- **Sometimes not as user friendly as JavaScript solutions**

```
<input type=text required name=foo>
```

pattern

- Using the pattern attribute, you can declare your own requirements for validation using Regular Expressions (RegEx).
- Implies validation. The required attribute only needed if input can't be blank.
- Like all validated fields, must have a name attribute.

```
<input type=text name=pattern-test
    pattern="[0-9][A-Z]{3}">
```

accept

- The accept attribute is supposed to limit the file selection dialog to files with certain MIME types.
- Types: audio/*, video/*, image/*, or other valid MIME types with no parameters.
- Fallback is that input accepts files of any type.

```
<input type=file accept="image/*">
```

placeholder

- Places text inside the input in a light gray color. The text remains whenever the input has no value. As soon as the input takes on value (from typing or any other means), the placeholder text is removed.

- Not implemented on textarea for all browsers

- Placeholders should be treated like a "hint." They are not a replacement for **labels** (accessibility) or **titles** (longer advisory text) or **value**

```
<input type=text placeholder="(555) 555-5555">
```

autofocus

- The first input in source order that has the autofocus attribute will be focused on page load.
- In browsers without autofocus support, no fields will be focused on page load.

<input autofocus type=text>

autocomplete

- Can prevent a specific field from being auto-filled.
- No visual difference between supported and unsupported browsers.
- Values are "on" and "off”
- Possibly useful for extra security on password fields

```
<form autocomplete=off>
```

vs

```
<input type=text name=name
autocomplete=off>
```

novalidate

- Used to turn off validation for a form, despite the attributes of the inputs it contains (i.e. will override inputs with the required attribute, or that would otherwise fail validation).

- Forms naturally validate, no special attributes needed. This special attribute is needed to turn that off.

```
<form novalidate>
        <input type=number name=num step=5>
</form>
```

Still to come

- Form validation

UNIVERSITY OF MICHIGAN

Credits: