Form Basics

Lesson 15

FEWD

GENERAL ASSEMBLY

Learning Objectives

- Differentiate between the different form input types, and explain why/where we might use each.
- Explain how to group different form elements by name.
- Apply the method, action, and enctype attributes to the form element.

Agenda

- HTML Forms
- Form Elements
- Input Attributes
- Styling & Validation
- Lab Time

HTML Forms

What is a Form?

A **form** in HTML is essentially just a wrapper for data collection elements that tells the page:

- Where to send the data
- How to send the data
- What data is being sent

HTML Form Syntax

```
<form action="" method="" enctype="">

    <!-- Form Input tags go here ... -->

</form>
```

Form Tag Attributes

- **Action** (url to send data to)
- **Method** (how the data should be sent)
- **Enctype** (the type of encoding for the data)

Action Attribute

The **action** attribute defines where the data gets sent. Its value must be a valid URL -- relative or absolute.

This will typically be a file on your server that parses it and does something with it.

```
<form action="my_file.php">
```

Method Attribute

The **method** attribute defines how data from a form is sent to the server.

The most common methods to send data via an HTML form are either **GET** or **POST**

```
<form method="get">
```

GET vs POST

- **GET** - form data is appended to the URL as a series of name/value pairs when submitted
- **POST** - form data is included in the request body instead of the URL
- GET vs POST

Enctype Attribute

The **enctype** attribute specifies how the form data should be encoded when submitting it to the server.

```
<form enctype="multipart/form-data">
```

Enctype Values

- **multipart/form-data** - No characters are encoded. This value is required when your form accepts file uploads
- **application/x-www-form-urlencoded** - (default) All text characters are encoded before being sent to the server

Pulse Check

Key Objective:

Let's explore these 2 different forms: **Msiddeeq GET** and **Msiddeeq POST**, and discuss the similarities and differences.

Timing:

- **10 minutes** - As a class, let's explore the differences of how forms are submitted

Form Elements

Form Elements

HTML **form elements** are what we place in between the open and close form tags to capture the user's input:

- Input**
- Select
- Textarea
- Label
- and more...

Code Along

Open: **Form Elements CodePen**

Select & Option

The **select element** creates a dropdown list. And we use **option elements** to populate the dropdown list with values

```
<form>
    <select>
        <option>Select an option...</option>
        <option value="val1">Value 1</option>
        <option value="val2">Value 2</option>
        <option value="val3">Value 3</option>
    </select>
</form>
```

Textarea

The **textarea element** allows you to type in multi-line text data.

```
<form>
    <textarea rows="10" cols="100"></textarea>
</form>
```

Label

The **label element** simply defines a label for another form element.

To tie the two together, you can choose one of these methods:

```
<form>
    <label for="first-name">First Name:</label>
    <input type="text" id="first-name" name="fname">

    <label>Last Name: <input type="text" name="lname"></label>
</form>
```

Input

The **input element** specifies an input field where the user can enter data.

An input field can vary in many ways, depending on the "type" attribute.

```
<form>
    <input type="text" name="fname">
    <input type="email" name="email">
    <input type="password" name="password">
</form>
```

Input Attributes

Input Attribute Examples

- Type**
- Value
- Name
- Placeholder
- and more...

Code Along

Open: **Form Elements CodePen**

Attribute: Name

The **name** attribute specifies the name of a form element.

It's used to reference elements in JavaScript, or to reference form data after a form is submitted.

```
<input type="text" name="fname">
```

IMPORTANT

REMEMBER: Only form elements with a name attribute will have their values sent to the server when submitting a form.

Attribute: Placeholder

The **placeholder** attribute is a short hint for what to type in the input field that displays inline in the input field itself

```
<input type="text" placeholder="Please enter your first name">
```

Attribute: Value

The **value** attribute specifies the value of an input element.

Depending on the field type, the value attribute can behave differently.

```
<input type="text" value="Mansoor">
```

IMPORTANT!

REMEMBER: It behaves differently depending on the input type:

- For **button**, **reset**, and **submit** - it defines the text on the button
- For **text** and **password** - it defines the initial (default) value of the input field
- For **checkbox** and **radio** - it defines the value associated with the input (this is also the value that is sent on submit)

Input Type Attributes

The **type** attribute specifies the type of <input> element to display.

The default type is text.

```
<input type="text">
```

Type: Text

This is the default. It defines a single-line text field.

It will accept any textual characters as input.

```
<input type="text">
```

Type: Email

This defines a field for an email address.

The input value is automatically validated to ensure it is a properly formatted e-mail address.

```
<input type="email">
```

Type: Password

This defines a field for a password.

The characters are masked when typing into the field.

```
<input type="password">
```

Type: Number

This defines a field for entering a number.

You can also set restrictions on what numbers are allowed using special attributes.

```
<input type="number">
```

Type: Radio

This defines a radio button.

Radio buttons are generally used in radio groups which allow the user to **select 1 option** of a limited number of choices.

Type: Radio (cont'd)

To accomplish this, all of the related radio buttons must have the same **name** attribute and different **value** attributes:

```
<input type="radio" name="car" value="Toyota">
<input type="radio" name="car" value="Honda">
<input type="radio" name="car" value="Ford">
```

Type: Checkbox

This defines a checkbox.

Similar to radio buttons, checkboxes can be used in groups. However, the user can select **multiple options**.

Type: Checkbox (cont'd)

Again, all of the related radio buttons must have the same **name** attribute and different **value** attributes:

```
<input type="checkbox" name="car" value="Toyota">
<input type="checkbox" name="car" value="Honda">
<input type="checkbox" name="car" value="Ford">
```

More Input Types...

- **type="file"** creates a file upload element
- **type="submit"** creates a clickable button that submits the form when clicked on
- **type="button"** creates clickable button

Break time!

Let's take 5-10 minutes to decompress...

Styling and Validation

Styling my Form Elements

Generally speaking, it is pretty simple to style basic text fields, buttons, etc.

However, it can be more difficult when attempting to style other elements such as select menus, file/date pickers, etc.

Check out this article for more details: **Styling HTML forms**

Form Validation

When accepting any user input on your site, you should **ALWAYS** validate it. Always.

Form Validation Methods

There are several methods for validating user input:

- Simple if/else statements or Regular Expressions
  - Regexr.com
  - Common Regular Expressions in JS
- Javascript libraries (e.g. Parsley.js)
- Server-side scripts

Lab Time

Open: **starter_code/forms**

Directions:

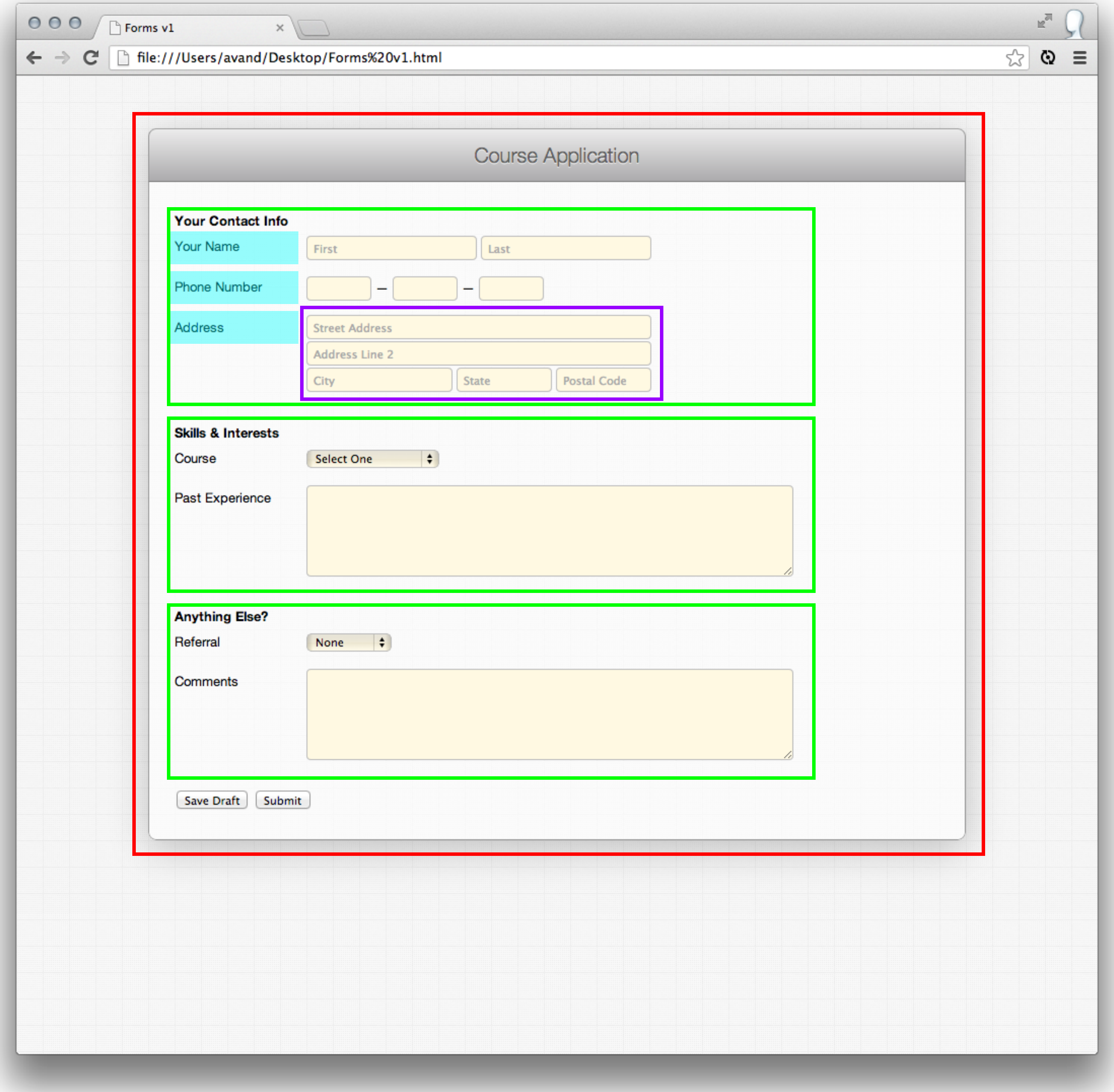
1. Open the **starter_code/forms** folder
2. Create the requisite HTML and CSS files/folders
3. Use the images in the screenshot folder as your guide
4. Build the form using HTML/CSS

Timing:

- **70 minutes** - build out the HTML/CSS for the form based on the screenshot images

Tips:

- Use fieldsets and/or divs to group your form elements/labels
- Remember, form elements are inline by default (you may need to change this to get it to work properly)



Course Application

Your Contact Info

Your Name: First, Last

Phone Number: ___ – ___ – ___

Address: Street Address, Address Line 2, City, State, Postal Code

Skills & Interests

Course: Select One

Past Experience

Anything Else?

Referral: None

Comments

Exit Tickets

Take 5-10 minutes to give us some

(Link is in Slack Room)

Learning Objectives Review

- We differentiated between the various form input types, and explained why/where we might use each.
- We explained how to group different form elements by name.
- We applied the method, action, and enctype attributes to the form element.

Week 7 HW Due Tonight

- **Due tonight by 11:59pm ET**
- Once you upload your HW to your repo, please PM the TA with a link to the Assignment folder.
- This is how we will know whether an assignment has been completed or not
- Once graded, the TA will reply with feedback

Final Projects

- **Milestone 3 (due tonight):** First draft of JS
- **Milestone 4 (due Monday, June 24th):** Rough Draft of Final Project

Next Class...

Lesson 16 - Forms "To-do List" Lab