CS 110

Lab 6: HTML Forms and Buttons

Overview

In this lab, you will create an interactive pizza ordering form using HTML forms, CSS styling, and JavaScript (You are given the JavaScript code, you only need to know how to use it. We will learn a bit about JavaScript at the end of the term). The lab has two main components: a pizza order form and a results page displaying the submitted order information.

Getting Started

Download these starter files from Canvas and save them to your lab6 folder:

- pizza_starter.html: Basic HTML structure with incomplete form elements (Get rid of the _starter part from the name after you download it.)

- pizza_styles_starter.css: CSS file with partial styling (Get rid of the _starter part after you download it so that href in pizza.html works)

- results_starter.html: Incomplete order confirmation page (Get rid of the _starter part from the name after you download it.)

- results-styles.css: Styling for the confirmation page

Understanding HTML Form Elements

Form Controls and Attributes

1. Label Element (`<label>`)
   - Purpose: Creates accessible, clickable captions for form controls
   - Key attribute: `for="input-id"`
   - Example: `<label for="user-name">Name:</label>`
2. Input Element (`<input>`)
   - Purpose: Creates interactive form controls
   - Key attributes:
     - `type`: Input behavior (text, tel, email, radio, checkbox)
     - `name`: Identifies data in form submission
     - `id`: Connects input to label
     - `required`: Makes field mandatory
     - `value`: Sets default value

3. Common Input Types
   o text: Basic text input
   o tel: Phone numbers
   o email: Email with validation
   o radio: Single selection
   o checkbox: Multiple selection
   o submit: Form submission button
   o reset: Clear form button

Part 1: Pizza Order Form

Step 1: Complete Customer Information Section

In pizza_starter.html, locate the customer information fieldset. Add the following form elements:

1. Complete the telephone input:

```
<li><label for="form-tel">Telephone Number:</label>
    <input type="tel" name="telephone" id="form-tel" required></li>
```

2. Add the email input:

```
<li><label for="form-email">Email:</label>
    <input type="email" name="email" id="form-email" required></li>
```

3. Complete the delivery instructions textarea (already partially provided)

Step 2: Complete Pizza Crust Options

In the crust selection fieldset, add the remaining radio button options:

```
<li><label><input type="radio" name="crust" value="multigrain"> Multigrain</label></li>
<li><label><input type="radio" name="crust" value="cheese-stuffed"> Cheese-stuffed</label></li>
<li><label><input type="radio" name="crust" value="gluten-free"> Gluten-free</label></li>
```

Step 3: Add Pizza Toppings

Complete the toppings fieldset by adding these checkbox options:

```
<li><label><input type="checkbox" name="toppings" value="pepperoni"> Pepperoni</label></li>
<li><label><input type="checkbox" name="toppings" value="mushrooms"> Mushrooms</label></li>
<li><label><input type="checkbox" name="toppings" value="peppers"> Peppers</label></li>
<li><label><input type="checkbox" name="toppings" value="anchovies"> Anchovies</label></li>
```

Step 4: Complete Pizza Quantity Selection

Add the remaining options to the quantity select element:

```html
<option>3</option>
<option>4</option>
<option>5</option>
<option>6</option>
```

Step 5: Add Form Submission Buttons

Add the submit and reset buttons at the bottom of the form:

```html
<input type="submit" value="Bring me a pizza!">
<input type="reset">
```

Understanding Form Buttons

- Submit Button (`type="submit"`)

    - Creates a clickable button that submits the form

    - `value` attribute sets the button's text

    - Triggers form validation

    - Sends form data to URL specified in form's `action` attribute

    - Style differs from text inputs - appears as a button instead of text field

    - Browser provides default button styling

- Reset Button (`type="reset"`)
    - Creates a button that clears all form inputs
    - Returns all fields to default values
    - No `value` attribute needed - uses browser's default "Reset" text
    - Does not submit form or validate
    - Useful for starting over
    - Also appears as button rather than text field

Buttons vs. Text Inputs:

    - Visual appearance: Rendered as clickable buttons

    - Interaction: Click triggers action instead of accepting text

    - Default styling: Browser provides button-specific styles

    - Purpose: Perform form actions rather than collect data

In pizza_styles_starter.css, add the following style rules:

4. Form input styling:

```css
input[type="text"],
input[type="tel"],
input[type="email"],
textarea {
    width: 300px;
    margin: 0.5em 0;
}
```

This selector group:

- Targets all text-based inputs and textareas

- Creates uniform width for visual consistency

- Adds vertical margins for better spacing

- Applies to telephone, email, and regular text inputs

2. Label alignment:

```css
label {
    display: inline-block;
    width: 150px;
    vertical-align: top;
}
```

This styling:

- Makes labels behave like blocks while remaining inline

- Creates consistent label column width

- Properly aligns labels with multi-line inputs

- Ensures uniform form layout

3. Button styling:

```css
input[type="submit"],
input[type="reset"] {
    margin: 1em 1em 1em 0;
    padding: 0.5em 1em;
}
```

This styling:

- Adds space between buttons (right margin)

- Creates vertical spacing around button group

- Makes buttons more clickable with padding

- Maintains consistent spacing with form elements

Part 2: Results Page

Understanding the JavaScript

The results_starter.html file contains a script that processes form submissions. This script uses the URLSearchParams API to extract form data from the URL query string when the page loads. It contains a helper function displayParam() that safely retrieves and displays parameter values in designated HTML elements. The script automatically populates customer information and pizza specifications **by matching parameter names with element IDs**.

Note: We will learn more about JavaScript later in the term, you only need to know what it is doing for now.

Step 7: Complete Results Page HTML

In results_starter.html, add:

1. Delivery instructions display after the customer information section:

```
<p><strong>Delivery instructions:</strong> <span id="instructions"></span></p>
```

2. Pizza order details section:

```
<ul>
    <li><strong>Crust:</strong> <span id="crust"></span></li>
    <li><strong>Toppings:</strong> <span id="toppings"></span></li>
    <li><strong>Number:</strong> <span id="pizzas"></span></li>
</ul>
```

Testing

Form Controls Testing

1. Required Fields
   o Test submission without filling required fields
   o Verify email format validation
   o Check telephone number input
   o Test maximum length on delivery instructions

2. Form Controls
- o Verify radio buttons allow only one selection
- o Test checkbox multiple selections
- o Confirm dropdown shows all quantity options
- o Test reset button clears all inputs

Results Page Testing
1. Complete Order Flow
- o Fill out the pizza order form completely
- o Submit and verify all information displays correctly
- o Check that delivery instructions appear
- o Verify multiple toppings are comma-separated
- o Confirm empty fields show "empty" as their value
2. Responsive Design
- o Test both pages at different screen sizes
- o Verify all elements remain usable on mobile
- o Check text remains readable at all sizes

Submission Requirements

1. Validate all HTML files using the W3C Validator
2. Validate CSS files using the W3C CSS Validator
3. Test complete order flow from form to results
4. Submit:
- o completed pizza.html
- o completed pizza-styles.css
- o completed results.html

Grading Criteria

- Working form controls (25%)
- Functional results page (25%)
- Proper HTML structure and validation (20%)
- CSS styling implementation (20%)
- Code organization and comments (10%)