

HTML Forms

Objectives

- View the HTML used to create forms
- Create a web form
- View some typical styling for web forms.

Forms

- Web forms are used to capture input from users.
- Web forms have specific HTML elements associated with them.
- These elements are standard across browsers.
- The forms themselves don't include any styling and are rather ugly.

Forms Best Practices #1

- There are various best practices around usability that apply to forms.
- Certain input types are better suited to capturing information... more on this later.
- CSS can be used to style forms to obtain a nicer look and feel.

The Form Element

- At the heart of web forms is the **form** element.
- The `<form>` element uses several attributes:
 - **action** – Where the form data will be sent
 - **method** – What HTTP method will be used
 - Optionally, **name** and **id** to help identify it for JavaScript.

Forms Need Input

- Forms wouldn't be much fun without input.
- The **input** element, which comes in several forms, captures input from users.
- Beyond the input element there are also other elements such as **<textarea>** and **<select>** that gather input on forms.

Input Types

- Aside from `<select>` and `<textarea>`, the types of inputs that correspond to `<input>` include:
 - text – A text box for free-form input
 - password – A text box that masks the input.
 - checkbox – Boxes that enable the user to select multiple values
 - radio – Small circles that enable the user to select one option from multiple
 - submit/image/reset/button – The various "button"-like inputs that are used to submit a form or reset it.

Forms Practice

- Create a new web page with a form.
- The form should contain two text inputs.

```
<form action="#" method="POST">
```

```
Your Username: <input type="text" name="username"><br>
```

```
Password: <input type="password" name="password"><br>
```

```
<input type="submit" name="submit" value="Go">
```

```
</form>
```


Divide and Conquer

- Forms can use the `<fieldset>` element to provide for logical groupings.
- The `<legend>` provides a friendly name for that portion of the form.
- The fieldset wraps around the inputs and other elements with opening and closing tags.
- The legend appears just after the fieldset to provide the section name.

Practice with Fieldset and Legend

- Add a fieldset and legend to your code.

<fieldset>

<legend>Login</legend>

Your Username: <input type="text"
name="username">

Password: <input type="password" name="password">

<input type="submit" name="submit" value="Go">

</fieldset>

Labels

- A form that is well-designed uses the `<label>` element to provide clear labels for each input.
- The `<label>` can be used by assistive technologies to help people use your form.
- Labels are connected to inputs with the "for" attribute:
- `<label for="username">Username: </label>`
- Labels typically require the use of an "id" on the input type.

Labels Practice

```
<fieldset>
```

```
<legend>Login</legend>
```

```
<label for="username">Your Username: </label>
```

```
<input id="username" type="text" name="username"><br>
```

```
<label for="password">Password:</label>
```

```
<input id="password" type="password"  
name="password"><br>
```

```
<input type="submit" name="submit" value="Go">
```

```
</fieldset>
```

Radio Buttons

- A group of radio buttons `<input type="radio">` share a common "name" attribute but use the "value" attribute to differentiate between which one was selected.
- Radio button groups are good for two to several (up to 10?) options. For more, use a `<select>` element.
- `<input type="radio" name="crust" id="deep" value="deep"> Deep Dish`
- `<input type="radio" name="crust" id="thin" value="thin"> Thin Crust`

Checkboxes

- Checkboxes enable multiple-value selection.
- `<input type="checkbox">`
- Like radio buttons, a group of checkboxes shares the same name.
- Checkboxes are appropriate for two to several (5 or 6?) selections but not many more than that.
- If you need more options, then a `<select>` box is a better option.

Select/Dropdown

- The `<select>` element creates a dropdown with multiple values.
- Select is appropriate for two to many (100?) options.
- You can configure the select box to allow one selection (like a radio button) or multiple (like a checkbox).

Select Syntax

- Note the syntax difference between select and other inputs.

```
<select name="size">
```

```
  <option value="small">Small</option>
```

```
  <option value="medium">Medium</option>
```

```
  <option value="large">Large</option>
```

```
</select>
```

- Add the word "multiple" to the <select> element to allow multiple selections.

Other Inputs Practice

- Add a radio button group (with two+ options), a checkbox group (with two+ options), and a select menu (with several options) to your form.
- Reminders:

```
<input type="radio" name="crust" value="deep">
```

```
<input type="checkbox" name="toppings"  
value="onions">
```

```
<select name="size"><option value="small">Small</  
option</select>
```

Text Area

- You can gather large areas of text input with the `<textarea>` element.
- This element is frequently used to obtain message-like input such as on a contact form.
- `<textarea></textarea>`

Form Submission and Buttons

- Forms are submitted with an input type of submit.
- You can also have an image or button in place of the standard submit button.
- There is also an input type of "reset" which resets the entire form (clears it).
- `<input type="submit" name="submit">`
- `<input type="reset" name="reset">`

Placeholders and Focus

- The placeholder attribute provides a hint for the user.
- `<input type="text" name="email" id="email" placeholder="Your e-mail address">`
- You can also set the focus where you'd like the user to begin with the autofocus attribute.
- `<input type="text" name="username" id="username" autofocus>`

Styling Forms

- Styling web forms can be rather challenging.
- The basic best practice is to align fields to be visually appealing while also making it clear to the user which labels correspond to which fields.
- There are no specific CSS properties for styling forms.
- However, you can use selectors to get at certain types of inputs.

CSS Selectors for Inputs

- Consider this CSS:

```
input[type="text"] {  
  margin-left: 1em;  
  margin-bottom: .5em;  
}
```

- Change "type" to "radio" or "checkbox" to work with those.

Alignment Practice

- Add an external stylesheet to your HTML and add the following CSS to it:

```
label {  
    width: 120px;  
    text-align: right;  
    display: inline-block;  
}  
  
input[type="text"] {  
    margin-left: 1em;  
    margin-bottom: .5em;  
}
```

CSS Styling for Usability

- Add a larger border to the input element that has focus:
 - `input:focus { border: 2px solid green; }`
- Change the background color:
 - `input:focus { background: grey; }`
- Add a box-shadow to the submit button:
 - `input[type="submit"] { box-shadow: 2px 2px 0 blue; }`

Summary

- Forms are the primary means to gather input on the web.
- The basic `<form>` element is combined with `<input>` elements (and `<select>` and `<textarea>`) to create a web form.
- The form would be submitted to a server-side program and typically validated by JavaScript.
- Validation will be covered next...