

---

---

# Building Forms

AIM CODE SCHOOL  
By: Vanessa Kasun

---

---

# Intro to Forms

Forms are an essential part of the internet as they allow websites to capture info from users and to process request. They also offer controls for nearly every imaginable use of an application.

In this lesson we will discuss how to use HTML to capture types of data and how to styles forms with CSS.

---

---

# Initializing a Form

In order to add a form to a page, we'll use the **<form>** element.

This element identifies where on the page control elements will appear. In addition the **<form>** element will wrap all of the elements included within the form, much like a **<div>** element.

```
<form action="/login" method="post">  
  ...  
</form>
```

---

# Text Fields

One of the main elements used to obtain text from users is the `<input>` element. The ***type*** attribute is used with the `<input>` element as well as the ***name*** attribute.

The ***type*** attribute is used to define what *type* of information is to be captured within the control. The most commonly used value is the ***text***, which denotes a single line of text input.

The ***name*** attribute is used as the name of the control and is submitted along with the input data to the server.

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

# More Attributes

These values provide clearer semantic meaning for inputs as well as to provide better controls for users. Should a browser not understand one of these HTML attribute values, it will automatically fall back to the text attribute value. Here is a list of new HTML5 input types.

- color
- email
- range
- time
- date
- month
- search
- url
- datetime
- number
- tel
- week

# Textarea

The `<textarea>` element is different from the `<input>` element in that it can accept larger passages of text spanning multiple lines. It also has start and end tags that can wrap plain text. The `<textarea>` tag only accepts only one type of value, because of this the **type** attribute doesn't apply here, but the **name** attribute does.

```
<textarea name="comment">This is the textarea tag </textarea>
```



This is an input tag

This is the textarea  
tag

# Multiple Choice Inputs & Menus

Aside from text-based input controls, HTML also allows users to select data using multiple choice and drop down lists.

## Radio Buttons

Radio buttons are usually presented in radio groups (a collection of radio buttons describing a set of related options). Only one radio button in a group can be selected at the same time.

```
<input type="radio" name="employment" value="Full-Time" checked> Full Time  
<input type="radio" name="employment" value="Part-Time"> Part Time  
<input type="radio" name="employment" value="Temporary"> Temporary
```

☒ Full Time ☐ Part Time ☐ Temporary

# Check Boxes

Radio buttons and checkboxes are very similar as they use the same attributes and patterns with the exception of checkbox as their type attribute value.

The difference between checkbox and radio is that checkboxes let users select a variety of values and tie them all to one control name, while radio buttons limit users to one value.

```
<input type="checkbox" name="month" value="January" checked> January  
<input type="checkbox" name="month" value="February"> February  
<input type="checkbox" name="month" value="March"> March|
```

☒ January ☐ February ☐ March



# Drop Down Lists

The `<select>` tag is used to create a **drop-down-list** with the `<option>` tag. The `<select>` element is most often used in a form, to collect user input. The ***name*** attribute is needed to reference the form data after the form is submitted. If you forget to add the name attribute, no data will be submitted. As well, the ***id*** attribute is need to associate the list with a label.

## Drop Down List Attributes

1. **Autofocus:** specifies that the drop down list should automatically get focus when the page loads
2. **Disable:** specifies that a drop down list should be disabled.
3. **Form:** defines which form the drop down list belongs to
4. **Multiple:** specifies that multiple options can be selected at once
5. **Name:** defines a name for the drop down list
6. **Required:** specifies that the user is required to select a value before submitting the form
7. **Size:** defines the number of visible options in a drop down list

```
<!DOCTYPE html>
<html>
<body>

<h1>The optgroup element</h1>

<p>The optgroup tag is used to group
  related options in a drop-down list:</p>

<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select name="cars" id="cars">
    <optgroup label="Swedish Cars">
      <option value="volvo">Volvo</option>
      <option value="saab">Saab</option>
    </optgroup>
    <optgroup label="German Cars">
      <option value="mercedes">Mercedes</option>
      <option value="audi">Audi</option>
    </optgroup>
  </select>
  <br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

# The optgroup element

The optgroup tag is used to group related options in a drop-down list:

Choose a car:

# Placeholder

This attribute provides a hint or top within the form control of an `<input>` or `<textarea>` element that disappears once the control is clicked in or gains focus.


```
<label>  
  Email Address  
  <input type="email" name="email-address" placeholder="name@domain.com">  
</label>
```

Email Address

# Submit Button

The `<input type="submit">` defines a submit button which submits all **form** values to a **form-handler**. The **form-handler** is typically a server page with a script for processing the input data. The **form-handler** is specified in the form's **action** attribute.

```
<button name="submit">  
  <strong>Send Us</strong> a Message  
</button>
```

A rectangular button with rounded corners, a light gray background, and a thin gray border. The text "Send Us a Message" is centered on the button in a bold, black, sans-serif font.

**Send Us a Message**

---

# Organizing Form Elements

Knowing how to capture data with inputs is half the battle. Organizing form elements and controls in a usable manner is the other half. When interacting with forms, users need to understand what is being asked of them and how to provide the requested information.

By using labels, fieldsets, and legends, we can better organize forms and guide users to properly complete them.

---

# Label

Labels may include a **for** attribute. The value of the **for** attribute should be the same as the value of the **id** attribute on the form control the label corresponds to. Matching the two together ties them together allowing users to click on the **<label>** tag to bring focus to the proper form control.

```
<label for="username">Username</label>  
<input type="text" name="username" id="username">
```

Username

# Fieldsets

The `<fieldset>` element is used to group several controls as well as labels (`<label>`) within the form.

```
<fieldset>
  <label>
    Username
    <input type="text" name="username">
  </label>
  <label>
    Password
    <input type="text" name="password">
  </label>
</fieldset>
```

Username

Password

# Legend

The `<legend>` element represents a caption for the content of its parent `<fieldset>`

```
<fieldset>
  <legend>Login</legend>
  <label>
    Username
    <input type="text" name="username">
  </label>
  <label>
    Password
    <input type="text" name="password">
  </label>
</fieldset>
```

Login

Username

Password



# Login Form Example

[Visit this link to see a login form example](#)