



# HTML and CSS

8th Edition

## Chapter 16: Forms



# Objectives

- Create, process, and validate forms.
- Create text boxes, password boxes, and text areas.
- Make email, search, telephone, and URL boxes.
- Use radio buttons, checkboxes, and submit buttons.
- Create select boxes.
- Allow visitors to upload files.
- Create hidden fields.
- Disabling form elements.



# Forms

- Constructing a form is straightforward and similar to creating any other part of a webpage.
- There are two basic parts of a form:
  - **The collection of controls, labels, and buttons** that the visitor views on a page and hopefully fills out or activates.
  - **The processing script** takes that information and converts it into a format that you can read or tally.
- Some of the form controls include text boxes, radio buttons, checkboxes, drop-down menus, and larger text areas.



# Creating Forms

- **Each form includes:**
  - Start and end tags.
  - Labels, controls, and buttons that make up the form.
  - Each control has a name attribute that identifies the data.
  - Visitors submit a form via a submit button.
  - The data is sent to the script on the server.
- The form start tag allows a few attributes:
  - **Action attribute:** Set the action attribute to the URL of the script that will process your form when it is submitted.
  - **Method attribute:** Has a value of either "get" or "post."



# To Create a Form

- Type `<form`  
`method="formmethod"`
- Type  
`action="script.url">`
- Create the form's contents.
- Type `</form>` to complete the form.

```
<body>
```

```
<h1>Create a New Account</h1>
```

```
<form method="post" action="show-data.php" > ...
```

```
</form>
```

```
</body>
```

## Create a New Account

**ACCOUNT**

First Name:

Last Name:

Email:

Password:

Re-enter Password:

**ADDRESS**

Street Address:

City:

State:

ZIP Code:

**PUBLIC PROFILE**

Picture:

Maximum size of 700K. JPG, GIF or PNG.

Screen Name:

Website URL:

Have a homepage or a blog? Put the address here, beginning with <http://> or <https://>.

Bio:

Gender:  Male  
 Female

**EMAILS**

It is okay to email me with messages from other users.

It is okay to email me with occasional promotions about our other products.



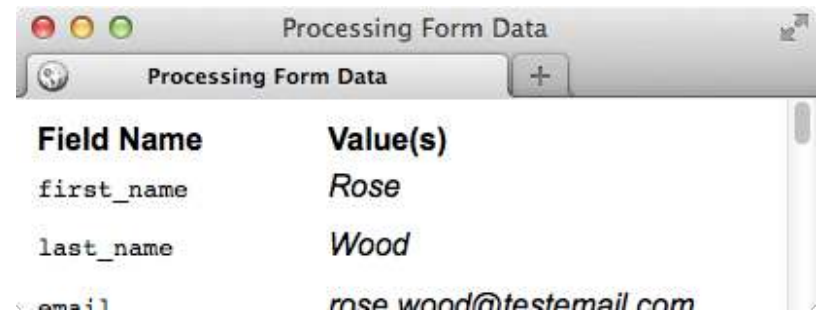
# Method="Get" and Method="Post"

- **method="get"** data will show in browser's address bar.
  - User can bookmark results.
  - Most search engines use method="get" in search forms so a search query can be saved or sent to a friend.
- **form method="post"** is for *posting* data on your server rather than getting data from it.
  - If you're saving, adding, and deleting data in a database, post is the correct choice.



# Processing Forms

- **Form scripts** process information gathered from visitors' forms.
- Processing scripts can:
  - Log info to database on server.
  - Send info via email.



The screenshot shows a browser window with the title 'Processing Form Data'. The window contains a table with two columns: 'Field Name' and 'Value(s)'. The table lists the following data:

Field Name	Value(s)
first_name	Rose
last_name	Wood
email	rose.wood@testemail.com



# Form Validation

- **To validate** a form means to check that the visitor has completed each necessary field and that the submitted data is in the format you expect.
  - Some form elements have built-in validation features.
  - These are not replacements for server-side validation, because older browsers or browsers with JavaScript disabled will not perform client-side validation.





# Server-Side vs. Client-Side

- **Server-side languages** runs on computer that serves your webpages, not visitor's computer.
  - Script must be uploaded to server to work.
  - PHP is a server-side language. Server must have PHP installed to interpret script.
- **Client-side languages** work inside browser.
  - Can do many tasks without interacting with the server at all.
  - Great for manipulating browser windows and checking data has been entered before submitting forms.
  - JavaScript is a client-side language.



# Organizing the Form Elements

- **Fieldset element** groups related elements in a form, making it easier to follow. Good if you have a lot of information.
- **Legend element** gives each fieldset a caption that describes the purpose of each grouping.

The screenshot shows a web browser window with the title "Creating a Form". The browser's address bar also displays "Creating a Form". The page content is organized into two distinct sections. The first section, titled "ADDRESS" in a blue header, contains four form elements: a text input for "Street Address:", a text input for "City:", a dropdown menu for "State:" with "Alabama" selected, and a text input for "ZIP Code:". The second section, titled "PUBLIC PROFILE" in a dark green header, is partially visible below the first section.

```
<form method="post" action="show-data.php">  
  <fieldset>  
    <h2 class="hdr-account">Account</h2>  
    ... Account fields ...  
  </fieldset>  
  
  <fieldset>  
    <h2 class="hdr-address">Address</h2>  
    ... Address fields ...  
  </fieldset>
```



# To Organize the Form Elements

- Type `<fieldset>`.
  - If desired, type `<legend>`.
- Type the text for the legend.
- Type `</legend>`.
- If you didn't include a legend, create a heading that identifies the group of form controls in the fieldset.
- Create form elements that belong in first group.
- Type `</fieldset>` to complete first group.
- Repeat steps for each group.



# Legend Element, Headings, Screen Readers, and Styling

- The legend element can help make your forms more accessible. Screen readers may announce the legend text for each form field associated with it, giving it additional context.
- This behavior varies by screen reader and browser, as well as by mode.
- Using an h1–h6 heading element to identify some (but not all) fieldsets can be a sensible alternative to using legend.



# Creating Text Boxes

- **Text boxes** can contain one line of freeform text entered by a visitor.
  - Typically used for things like names and addresses.
  - Each text box is represented by an input tag with `type="text"` assigned.
- **Name attribute** is used by server-side scripts to retrieve the value a visitor enters in the text box or that you prepopulate with the **value attribute**.
- Example: `<input type="text" name="firstName" />`
- **Note: use the value attribute if there is a default value.**



# To Create a Text Box

- Type label that identifies text box to visitor. `<label for="firstName">First Name</label>`
- Type `<input type="text">`.
- Type `name="dataname"`
  - `dataname` is text that identifies input data to server.
- If you created a label, type `id="idlabel"`
  - `idlabel` is matching text.
- Optionally, type `value="default"`
  - `default` is data that will initially appear in field.
- Optionally, type `placeholder="hinttext"`
  - `hinttext` is data initially shown in field as a hint to the user.



# To Create a Text Box

- Optionally, type *required* or *required="required"* and the form will not submit unless the field has a value.
- Optionally, type *autofocus* or *autofocus="autofocus"* to give the input element focus when the page loads.
- Optionally, to define size of form box, type *size="n"*
  - *n* is desired width of the box, measured in characters.
- Optionally, type *maxlength="n"*,
  - *n* is maximum number of characters that can be entered.
- Finish the text box by typing */>*.



# Textbox Attributes

## Placeholder

Creating a Form

ACCOUNT

First Name:

Last Name:

Email:

Password:

## Required

Creating a Form

ie:

Please fill out this field.

le.com

d:

d:





# Textbox Attributes

## Different sizes for fields

The screenshot shows a browser window titled "Creating a Form" with a tab labeled "Creating a Form". The main heading is "Create a New Account". Below this, there are two sections: "ACCOUNT" and "ADDRESS".

**ACCOUNT**

First Name:

Last Name:

Email:

Password:

Re-enter Password:

**ADDRESS**

Street Address:

City:

State:

ZIP Code:

Below the "ADDRESS" section, the start of a "PUBLIC PROFILE" section is visible.

## Autocomplete

The diagram shows a "City:" label followed by a text input field containing "Ma|". A dropdown menu is open below the input field, listing three options: "Manila", "Maplewood", and "Panama City". To the left of the dropdown, the labels "State:" and "ZIP Code:" are visible, indicating the context of the form fields.



# Labeling Form Parts

- **Label element** identifies text describing the purpose of a form field.
- **For attribute** is a special attribute associated with the label element. When for has the same value as a form field's id, the label and field are associated explicitly, improving the usability and accessibility of your forms.
- Example: `<label for="firstName">First Name</label>`
- When creating the form element, be sure to include an id that matches the for attribute.



# Naming Convention for Id, For, and Name Attributes

- Common but not required for the for, id, and name attributes to be identical.
  - Radio buttons and checkboxes are the exception.
- For multi-word values, consider separating each word with a hyphen ("-") in for and id and with an underscore ("\_") in name.



# Creating Password Boxes

- **Password box** is a text box where whatever is typed is hidden by bullets or asterisks.
  - Information is not encrypted when sent to server.

The screenshot shows a web browser window titled "Creating a Form" with a sub-window titled "ACCOUNT". The form contains the following fields:

- First Name: Rose
- Last Name: Wood
- Email: rose.wood@testemail.com
- Password: [masked with asterisks]
- Re-enter Password: [masked with asterisks]

```
...  
<p class="row">  
  <label for="password">Password:</label>  
  <input type="password" id="password"  
  → name="password" />  
</p>  
<p class="row">  
  <label for="password2">Re-enter  
  → Password:</label>  
  <input type="password" id="password2"  
  → name="password2" />  
</p>  
...
```



# To Create a Password Box

- Create a label to identify the password box.
- Type `<input type="password">`.
- Type `id="idlabel"`
  - `idlabel` is the same as the label's for attribute value in step 1.
- Type `name="dataname"`
  - `dataname` is the text that identifies input data to the server.
- If desired, define the form box's size by typing `size="n"`
  - `n` is the width of the box, measured in characters.



# To Create a Password Box

- If desired, type *maxlength="n"*
  - *n* is the maximum character count.
- If desired, type *required* or *required="required"*.
- If desired, type *autofocus* or *autofocus="autofocus"*.
- Finish the password box by typing `</>`.



# Creating Email, Search, Telephone, and URL Boxes

- **Email, telephone, and URL input types** are new to HTML5. They look exactly like text boxes, but have helpful features added for validating text.
  - Non-supporting browsers will treat the fields as normal text boxes without the extra features.

Creating a Form

ating a Form +

UNT

First Name:

Last Name:

Email:

Website URL:

Phone:

Please enter an email address.

Account



# Validating Field Text for URL

The screenshot shows a web browser window with a tab titled "Creating a Form". The page has a blue header with the text "UNT". Below the header, there are several form fields: "First Name:", "Last Name:", "Email:", "Website URL:", and "Phone:". The "Website URL:" field contains the text "www.wikipedia.org" and is highlighted with a red border. A tooltip message is displayed over the field, stating "Please enter a URL." Below the tooltip, there is a small text prompt: "Have a site or blog? Put the address here, e.g. http://www.example.com or https://www.example.com". At the bottom left of the form, there is an orange button labeled "Account".

- The browser checks to make sure that the Website URL field text is in the valid URL format.
- `www.wikipedia.org` is not a valid form URL, because a URL must begin with `http://` or `https://`.





# Validating Field Text for Phone

Creating a Form

ating a Form +

UNT

First Name:

Last Name:

Email:

Website URL:   
Have a site or a blog? Put the address here, beginning with http:// or https://.

Phone:

Please match the requested format.

Account

- When the visitor submits the form, the browser checks to make sure that the Phone field text (if any) matches the format specified in the pattern attribute.



# Creating Search Boxes

- **Search boxes** are also new in HTML5. They are just like text boxes, except some browsers make them appear like the default search boxes on their operating system.

Search:

```
<label for="search">Search:</label>  
<input type="search" id="search"  
→ name="search" size="30"  
→ placeholder="e.g., a book or  
→ magazine" />  
<input type="submit" value="Find It!" />
```



# Creating Search Boxes

- Non-supporting browsers will treat the fields as normal text boxes without the extra features.

Search:



# To Create Email, Search, Telephone, and URL Boxes

- Create a label to identify the input box to your visitors.
- Type one of the following:
  - `<input type="email">` for an email box
  - `<input type="search">` for a search box
  - `<input type="tel">` for a telephone number box
  - `<input type="url">` for a URL box
- Type `id="idlabel"`
  - `idlabel` is the same as the label's for attribute value.



# To Create Email, URL, and Telephone Boxes

- Type *name="dataname"*
  - *dataname* is the text that identifies input data to the server.
- If desired, type *value="default"*
  - *default* is the data initially shown in the field.
- If desired, type *placeholder="hinttext"*
  - *Hinttext* will initially show in the field as a hint to the user.
- If desired, type *required* or *required="required"*.



# To Create Email, URL, and Telephone Boxes

- If desired, type *autofocus* or *autofocus="autofocus"*.
- If desired, define size of the box by typing *size="n"*
  - *n* is the desired width of the box, measured in characters.
- If desired, type *maxlength="n"*
  - *n* is maximum number of characters that can be entered.
- Finish the text box by typing `/>`.



# Creating Radio Buttons

- **Radio buttons** are a set of buttons of which only one can be selected at a time.
- Create a radio button by setting `type="radio"` on an input.

The screenshot shows a browser window titled "Creating a Form". Below the title bar is a text input field labeled "io:". Below the text input field is a radio button group labeled "er:". The radio button group contains two options: "Male" and "Female". The "Female" radio button is selected, indicated by a blue dot and a mouse cursor hovering over it.



# To Create Radio Buttons

- If desired, type introductory text for radio buttons.
- Type `<input type="radio">`
- Type `name="radioset"`
  - `radioset` identifies data sent and links radio buttons.
- Type `id="idlabel"`
  - `idlabel` identifies unique radio button linked to label.
- Type `value="data"`
  - `data` is text sent to server if radio button is selected.





# To Create Radio Buttons

- If desired, type *checked* or *checked="checked"* to make the radio button active by default.
- Type `</>`.
- Type `<label for="idlabel">radio label</label>`
  - *idlabel* matches the id value in your radio button and *radio label* identifies the radio button to the visitor.
- Repeat for each radio button in the set.



# Radio Buttons Example

```
<h1>Display Radio Buttons</h1>

<form action="/action_page.php">
  <p>Please select your favorite Web language:</p>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language"
value="JavaScript">
  <label for="javascript">JavaScript</label>

  <br>

  <p>Please select your age:</p>
  <input type="radio" id="age1" name="age" value="30">
  <label for="age1">0 - 30</label><br>
  <input type="radio" id="age2" name="age" value="60">
  <label for="age2">31 - 60</label><br>
  <input type="radio" id="age3" name="age" value="100">
  <label for="age3">61 - 100</label><br><br>
  <input type="submit" value="Submit">
</form>
```

## Display Radio Buttons

Please select your favorite Web language:

- HTML
- CSS
- JavaScript

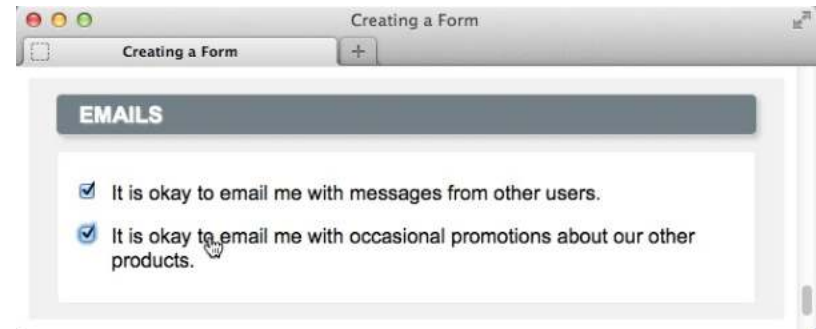
Please select your age:

- 0 - 30
- 31 - 60
- 61 - 100



# Creating Checkboxes

- **Checkboxes** are a set of boxes where a visitor can select as many options as they like.
- Like radio buttons, checkboxes in a set all have the same name attribute.





# To Create Checkboxes

- If desired, type the introductory text for checkboxes.
- Type `<input type="checkbox">`.
- Type `name="boxset"`
  - `boxset` identifies data sent to the server and also represents the checkboxes as a group. (Use the same name for all.)
- Type `id="idlabel"`
  - where `idlabel` matches the `for` attribute value of the label you'll create.
- Type `value="data"`
  - `data` is the text sent to the server if the checkbox is marked.



# To Create Checkboxes

- If desired, type *checked* or *checked="checked"* (either is fine) to make checkbox selected by default.
- Type */>* to complete the checkbox.
- Type *<label for="idlabel">checkbox label</label>*
  - *idlabel* matches the id value in your checkbox element, and *checkbox label* identifies the checkbox to the visitor.
- Repeat steps for each checkbox in the set.



# Checkboxes Example

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

<h1>Show Checkboxes</h1>

<form action="/action_page.php">
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> I have a boat</label><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

## Show Checkboxes

- I have a bike
- I have a car
- I have a boat



# Creating Text Areas

- **Text areas** give visitors more room to write questions or comments than text boxes.

```
<label for="bio">Bio:</label>  
<textarea id="bio" name="bio" cols="40"  
→ rows="5" class="field-large"></textarea>
```

Creating a Form

ne:

RL:

*Have a homepage or a blog? Put the address beginning with http:// or https://.*

Bio:



# To Create Text Areas

- Create a label to identify the text area to your visitors.
- Type `<textarea`.
- Type `id="idlabel"`
  - *idlabel* is the same as the label's for attribute value.
- Type `name="dataname"`
  - *dataname* is the text that identifies text area data to the server (and your script).
- If desired, type `maxlength="n"`
  - where *n* is the maximum number of characters that can be entered in the box.





# To Create Text Areas

- If desired, type `cols="n"`
  - *n* is roughly the number of characters visible per line (the text area's width).
- If desired, type `rows="n"`
  - *n* is the number of visible lines of text (the text area's height).
- Type `>`.
- Type the default text, if any, for the text area.
- Type `</textarea>` to complete the text area.



# Creating Select Boxes

- **Select boxes** offer visitors a choice from a given set of options.
  - Most often rendered as drop-down lists.
  - If there's an option to select multiple answers, it will render as a box of items with a scroll bar.

Street Address:

City:

State:

ZIP Code:

The State dropdown menu is open, showing a list of options: Alabama, Alaska, and California. The 'Alabama' option is highlighted in blue.



# To Create Select Boxes

- Create a label to describe your menu.
- Type `<select`.
- Type `id="idlabel"`
  - `idlabel` is the same as the label's for attribute value.
- Type `name="dataname"`
  - `dataname` will identify the data collected from the menu when it is sent to the server.
- If desired, type `size="n"`
  - `n` represents the height (in lines) of the select box.



# To Create Select Boxes

- If desired, type *multiple* or *multiple="multiple"* to allow your visitor to select more than one menu option (with the Control or Command key).
- Type `>`.
- Type `<option`.
- Type `value="optiondata"`
  - *optiondata* is the data that will be sent to the server if the option is selected.
- If desired, type *selected* or *selected="selected"* to specify that the option be selected by default.



# To Create Select Boxes

- Type `>`.
- Type the option text as you wish it to appear in the menu.
- Type `</option>`.
- Repeat steps for each option.
- Type `</select>`.



# Select Box Example

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

<h1>The select element</h1>

<p>The select element is used to create a drop-down list.</p>

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

<p>Click the "Submit" button and the form-data will be sent to a page
on the
server called "action_page.php".</p>

</body>
```

## The select element

The select element is used to create a drop-down list.

Choose a car:

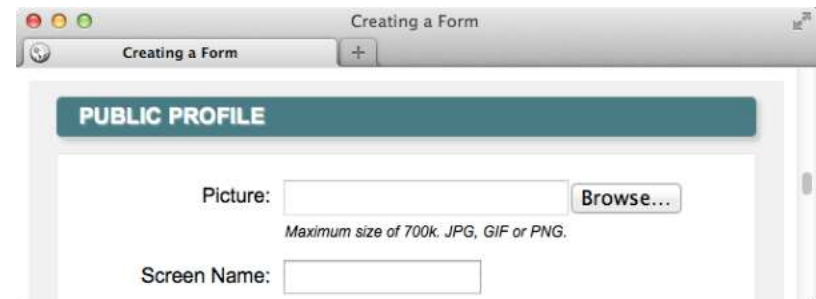
Click the "Submit" button and the form-data will be sent to a page on the server called "action\_page.php".

Volvo  
Saab  
Opel  
Audi



# Allowing Visitors to Upload Files

- You might want users to upload a file, such as a photo or résumé, to your server.
- The file upload area provides a way for the user to select a file on their system.



```
<label for="picture">Picture:</label>  
<input type="file" id="picture"  
→ name="picture" />  
<p class="instructions">Maximum size of  
→ 700k. JPG, GIF or PNG.</p>
```



# To Allow Visitors to Upload Files

- When you begin your form, type `<form method="post" enctype="multipart/form-data">`
- Type `action="upload.url">`
  - *upload.url* is the URL of script that processes incoming files.
- Create label for the file upload area so your visitors know what to upload.
- Type `<input type="file">` to create a file upload box and a Browse button.
- Type `id="idlabel"`
  - *idlabel* is the same value as the label's for attribute value.





# To Allow Visitors to Upload Files

- Type `name="dataname"`
  - *dataname* identifies the file or files being uploaded.
- If desired, type `size="n"`
  - *n* is width of field where visitor will enter path and file name.
- If desired, type `multiple` or `multiple="multiple"` to allow visitors to upload more than one file.
- Type the final `/>`.
- Complete the form as usual, including the submit button and `</form>` end tag.



# Creating Hidden Fields

- **Hidden fields** store data in a form without showing it to visitors. Think of them as invisible text boxes.
- Often used by processing scripts to store information gathered from an earlier form so that it can be combined with the present form's data.



# To Create Hidden Fields

- Type `<input type="hidden">`.
- Type `name="dataname"`
  - `dataname` identifies the information submitted to the server.
- Type `value="data"`
  - `data` is the information itself that is to be submitted. It is often a variable from the form processing script.
- Type the final `>`.



# Creating a Submit Button

- **Submit button** activates the script that sends the data to the server.
- Submit buttons may be text, an image, or a combination of both.





# To Create a Submit Button

- Type `<input type="submit">`.
- If desired, type `value="submit message"`.
  - *submit message* is the text that will appear in the button.
- Type the final `>`.





# To Create a Submit Button with an Image

- Create a PNG, GIF, or JPEG image.
- Type `<input type="image">`.
- Type `src="image.url"`.
  - `image.url` is the location of the image on the server.
- Type `alt="description"`
  - `description` is what will appear if the image does not.
- Type the final `>` to finish the image submit button.



# Using an Image to Submit a Form

- Browsers may show a hand instead of an arrow when you hover the cursor over an image submit button.





# Disabling Form Elements

- **Disable form element** disables parts of your form if you don't want visitors to use them.
- **To disable a form element:**
  - In the form element's start tag, type *disabled* or *disabled="disabled"*.





# Grayed Out Text Area

Text area is grayed out and disabled



**How did you hear about us?**

- Advertisement
- Email Newsletter
- FaceSpace
- Other (please explain if you select)

TV, school, bingo game, etc.

**SUBMIT**

Text area turns white and the user can enter text



**How did you hear about us?**

- Advertisement
- Email Newsletter
- FaceSpace
- Other (please explain if you select)

Read a magazine article on a train.

**SUBMIT**