

# Chapter 2

## HTML Part 5

### Forms

INT102 Web Technologies (1/2023)

Sanit Sirisawatvatana

# Topics

- What will you learn after finishing this class:
  - HTML Forms and form attributes
  - Form Controls
    - Text entry controls
    - Password field
    - Radio and Checkbox buttons
    - Drop-down and Scrolling menus
    - Submit and reset buttons
    - Hidden controls
    - Fieldset and Legend element

# HTML Forms

- HTML **Forms** are used to collect some information from the site visitors.
  - Ex. During user registration, user information such as name, email address, credit card are required.
- A **form** will take input from the site visitor and then will **post** it to a back-end application such as **CGI**, **ASP** Script or **PHP** script.
- The **back-end application** processes the information collected by the form and returns the appropriate **response** to the site visitor.

# From data entry to response

signup.html

## Mailing List Signup

Join our email list

Get news about the band such as tour dates and special MP3 releases sent to your own in-box.

1. Name:

2. Email:

username = Sanit  
email = sanit@sit.kmutt.ac.th

request



Web Application



mailinglist.php

response

## THANKS Sanit

sanit@sit.kmutt.ac.th are now on the band mailing list. Can't wait to see you out at the shows.

[Go back to the main page](#)

# signup.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mailing List Signup</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Mailing List Signup</h1>
    <form action="mailinglist.php" method="post">
      <fieldset>
        <legend>Join our email list</legend>
        <p>Get news about the band such as tour dates and special MP3
        releases sent to your own in-box.</p>
        <ol>
          <li>
            <label for="firstlast">Name:</label>
            <input type="text" name="username" id="firstlast">
          </li>
          <li>
            <label for="email">Email:</label>
            <input type="text" name="email" id="email">
          </li>
        </ol>
        <input type="submit" value="Submit">
      </fieldset>
    </form>
  </body>
</html>
```

## Mailing List Signup

Join our email list

Get news about the band such as tour dates and special MP3 releases sent to your own in-box.

1. Name:

2. Email:

[Demo](#)

# mailinglist.php

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Mailing List Signup Result</title>
  </head>
  <body>
    <hr>
    <h1>THANKS <?php echo $_POST["username"]; ?></h1>
    <hr>
    <p><?php echo $_POST["email"]; ?> are now on the band mailing list.
    Can't wait to see you out at the shows.</p>
    <a href="signup.html">Go back to the main page</a>
  </body>
</html>
```

---

**THANKS Sanit**

sanit@sit.kmutt.ac.th are now on the band mailing list. Can't wait to see you out at the shows.

[Go back to the main page](#)

# <form> element

- **Forms** are added to the web pages using the `<form>` element.
- The `<form>` element is **a container** for all the content of the form such as input field, buttons and drop-down menu (collectively known as **form controls**).
- The `<form>` element may contain block elements such as **p**, **h1**, and **lists**. However, it may not contain another form element.

# Form Attributes

Attribute	Description
-----------	-------------

action	The location ( <a href="#">URL</a> ) of the <a href="#">application</a> or <a href="#">script</a> (sometimes called the <a href="#">action page</a> ) that will be used to process the form
--------	---

method	The method to be used to upload data to the server: <a href="#">POST</a> or <a href="#">GET</a>
--------	---

`<form action="mailinglist.php" method="post">`



# Scripting Language

- PHP (.php)
- Microsoft's ASP.NET (.asp)
- Ruby on Rails
- JavaServer Pages (.jsp)
- Python

# method attribute

- The `method` attribute specifies how the information should be sent to the sever.
- Data gathered from the form  
`username = Sanit`  
`email = sanit@sit.kmutt.ac.th`
- Browser encodes that information for its trip to the server  
`username=Sanit&email=sanit%40sit.kmutt.ac.th`
- Two methods for sending this encoded data to the server:
  - POST
  - GET (`Default`)

# POST method

- The browser sends a server request containing some special headers followed by the data.
- Only the **server** sees the content of this request.
- The **POST** method is the best method for sending **secure information** such as credit card or other **personal information**.
- The **POST** method has **no size limitations** and can be used to **send a lot of data**.
- Form submissions with the **POST** method **cannot** be **bookmarked**.
- The **POST** method is used when the form is used to upload a file.

# GET method

- When the **GET** method is used, the submitted form data will be visible in the page address bar. The question mark character separates the URL from the following data:

<https://int102.sit.kmutt.ac.th/webtech140/maillinglist.php?username=Sanit&email=sanit%40sit.kmutt.ac.th>

- The length of a URL is **limited** (about 3,000 characters)
- The **GET** method allows users to be able to bookmark the results of a form submission.
- The **GET** method is not be used for the forms with private personal or financial information.

# Name Attribute

- The **name** attribute provides the **variable name** for the **control**.
- All form control elements must include a **name** attribute so the form-processing application can get the information.
- Each **variable** in a **form** must be named uniquely.

```
<h1>Mailing List Signup</h1>
<form action="mailinglist.php" method="post">
  <fieldset>
    <legend>Join our email list</legend>
    <p>Get news about the band such as tour dates and special MP3
    releases sent to your own in-box.</p>
    <ol>
    <li>
      <label for="firstlast">Name:</label>
      <input type="text" name="username" id="firstlast">
    </li>
    <li>
      <label for="email">Email:</label>
      <input type="text" name="email" id="email">
    </li>
    </ol>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

signup.html

```
<hr>
  <h1>THANKS <?php echo $_POST["username"]; ?></h1>
<hr>
<p><?php echo $_POST["email"]; ?> are now on the band mailing list.
Can't wait to see you out at the shows.</p>
```

mailinglist.php

# Form Controls

- There are different types of form controls that can be used to collect data using HTML form.
  - Text entry controls
  - Specialized text entry
  - controls Radio buttons
  - Checkbox buttons
  - Pull-down and scrolling menus
  - File selection and upload control
  - Hidden controls
  - Clickable buttons
  - Submit and reset buttons
  - Dates and times (HTML5)
  - Numerical controls (HTML5)
  - Color picker control (HTML5)

# <input> element

- The `<input>` element can be displayed in several ways, depending on the `type` attribute.

`<input type = "text">`

text

`<input type = "radio">`

☐ radio

`<input type = "checkbox">`

☐ check box

`<input type = "submit">`

# Single-line text field

`<input type = "text">`

- is used for entering a **single word** or line of **text**.
- It is the **default type** (when omitting the type attribute).
- Attributes
  - **name** defines the variable name
  - **value** specifies default text when the form is loaded.
  - **size** specifies the width of the text input control (characters)
  - **maxlength** specifies the maximum character limit.

<https://stackoverflow.com/questions/25247565/difference-between-maxlength-size-attribute-in-html>



# Text Field

<form >

First name: <input type = "text" name = "first\_name" >

<br>

Last name: <input type = "text" name = "last\_name" >

</form>



First name:

Last name:

[Demo](#)

`type="text"` → Specifies that the input field is for text input

`name` → Defines the name of the input field (used when submitting the form)

`id` → Associates the input with a `<label>` or can be used in JavaScript

`value` → Specifies the default value displayed in the input field

`placeholder` → Displays a hint text inside the input field

`required` → Makes the input field mandatory before form submission

`maxlength` → Limits the maximum number of characters allowed

`minlength` → Sets the minimum number of characters required

`size` → Defines the width of the input field in terms of character count

`readonly` → Prevents users from editing the input field

`disabled` → Disables the input field, making it non-interactive

`autocomplete` → Controls auto-fill suggestions (on / off)

`pattern` → Validates input using a regular expression

`autofocus` → Automatically focuses on the input field when the page loads

`spellcheck` → Enables or disables spell checking (true / false)

`inputmode` → Suggests a specific type of keyboard (text, numeric, email, tel)

`dir` → Sets the text direction (ltr for left-to-right, rtl for right-to-left)

# <label> element

- The <label> element defines a label for a <input>, <textarea>, <select> or <button> element.
- The <label> does not render as anything special for the user. It provides a usability improvement for mouse users.
- When the user clicks on the text within the <label> element, it toggles the control.
- The for attribute of the <label> element should be equal to the id attribute of the related element to bind them together.

# Text Field with Label

```
<form >
```

```
<label for ="first_name">First name: </label>
```

```
<input type = "text" name = "first_name" id="first_name" >
```

```
<br>
```

```
<label for ="last_name">Last name: </label>
```

```
<input type = "text" name = "last_name" id="last_name" >
```

```
</form>
```



First name:

Last name:

[Demo](#)

# Multiline text entry field

`<textarea>...</textarea>`

- This is used when the user is required to give details that may be longer than one line of text.
- Attributes
  - `name`
  - `rows` specifies the number of rows of text area box
  - `cols` specifies the number of columns of text area box
  - `placeholder` (HTML5) provides a short hint of how to fill the field.
- The `rows` and `cols` attributes are a way of specifying the size of the text area.
- Unlike the empty `input` element, the `content` can be put between the opening and closing tags in the `textarea` element.

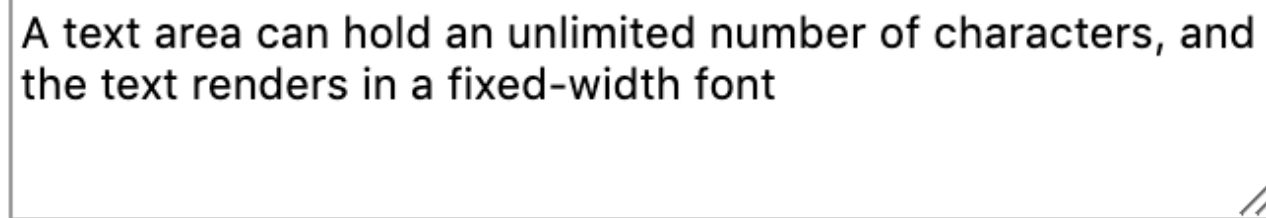
# <textarea> element

<form >

<textarea rows="4" cols="50"> A text area can hold an unlimited number of characters,  
and the text renders in a fixed-width font

</textarea>

</form>



A text area can hold an unlimited number of characters, and  
the text renders in a fixed-width font

[Demo](#)

- name → Defines the name of the textarea (used when submitting the form)
- id → Assigns a unique identifier for use with <label> or JavaScript
- rows → Specifies the number of visible text lines
- cols → Specifies the width of the textarea in terms of character count
- placeholder → Displays a hint text inside the textarea
- maxlength → Limits the maximum number of characters allowed
- minlength → Sets the minimum number of characters required (HTML5)
- required → Makes the field mandatory before form submission
- readonly → Prevents users from editing the content
- disabled → Disables the textarea, making it non-interactive
- wrap → Specifies how text should be wrapped (soft, hard, or off)
- autofocus → Automatically focuses on the textarea when the page loads
- spellcheck → Enables or disables spell checking (true / false)
- autocomplete → Controls auto-fill suggestions (on / off)
- form → Specifies which <form> the textarea belongs to
- dir → Sets the text direction (ltr for left-to-right, rtl for right-to-left)

# Specialized text entry fields

- Specialized text entry fields
  - Password entry field `<input type = "password">`
  - Search field (HTML5) `<input type = "search">`
  - Email address (HTML5) `<input type = "email">`
  - Telephone number (HTML5) `<input type = "tel">`
  - Location (URL) (HTML5) `<input type = "url">`

[https://www.w3schools.com/tags/tag\\_input.asp](https://www.w3schools.com/tags/tag_input.asp)



# Password entry field

`<input type = "password">`

- A **password** field works like a text entry field, except the characters are obscured from view using **asterisk** (\*) or **bullet** (•) or another character determined by the browser.
- The form does **not encrypt** the information, so it should not be considered a real security measure.

# Password field

<form>

<label for="form-pswd">Password</label>

<input type="password" name="pswd" maxlength="8" id="form-pswd">

</form>



[Demo](#)

# Radio and Checkbox buttons

- Both **radio** and **checkbox buttons** make it simple for visitors to choose from a number of provided **options**.
- **Radio buttons** are used when out of many options, just **one option** is required to be selected.
- **Checkboxes** are used when **more than one option** is required to be selected.

**<input type="radio">**

**Choose an Option:**

☐ Option 1 ☐ Option 2 ☐ Option 3

**HTML <input type="checkbox">**

Select Your Subjects:

- ☒ HTML
- ☐ CSS
- ☐ JavaScript

# Radio buttons

```
<input type = "radio" name = "variable" value = "value">
```

- The **name** attribute is required and play an important role in binding multiple radio inputs into set.
- When a number of radio button inputs are provided with the **same name value**, they create **a group of mutually exclusive options**.

# Radio buttons

```
<form>
  <p>Please select your preferred contact method:</p>
  <div>
    <input type="radio" id="contactChoice1"
      name="contact" value="email">
    <label for="contactChoice1">Email</label>

    <input type="radio" id="contactChoice2"
      name="contact" value="phone">
    <label for="contactChoice2">Phone</label>

    <input type="radio" id="contactChoice3"
      name="contact" value="mail">
    <label for="contactChoice3">Mail</label>
  </div>
  <div>
    <button type="submit">Submit</button>
  </div>
</form>
```

Please select your preferred contact method:

☐ Email ☐ Phone ☐ Mail

[Demo](#)

# Checkbox buttons

`<input type = "checkbox">`

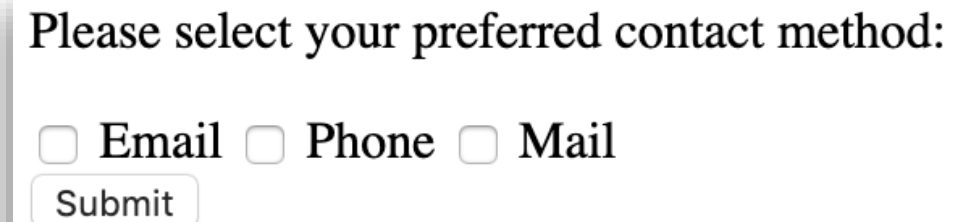
- As with radio buttons, **creating groups** of checkboxes is by assigning them the **same name value**.
- When more than one checkbox may be checked at a time, the value of every checked button will be sent to the server when the form is submitted.
- Checkboxes don't necessarily need to be used in groups.

# Checkbox buttons

```
<form>
  <p>Please select your preferred contact method:</p>
  <div>
    <input type="checkbox" id="contactChoice1"
      name="contact[]" value="email">
    <label for="contactChoice1">Email</label>

    <input type="checkbox" id="contactChoice2"
      name="contact[]" value="phone">
    <label for="contactChoice2">Phone</label>

    <input type="checkbox" id="contactChoice3"
      name="contact[]" value="mail">
    <label for="contactChoice3">Mail</label>
  </div>
  <div>
    <button type="submit">Submit</button>
  </div>
</form>
```



Please select your preferred contact method:

☐ Email ☐ Phone ☐ Mail

Submit

[Demo](#)

# Menus

```
<select name = "variable">  
  <option> choice 1 </option>  
  <option> choice 2 </option>  
  <option> choice 3 </option>  
</select>
```

- Menus provide a list of choices by putting them in a drop-down or scrolling menu.
- By default, the content of the chosen option is what gets passed to the web application when the form is submitted.
- To send a different value than what appears in the menu, use the value attribute to provide an overriding value.

```
<select name = "variable">  
  <option value = "1"> choice 1 </option>  
  <option value = "2"> choice 2 </option>  
  <option value = "3"> choice 3 </option>  
</select>
```



# Drop-down menus

- The `select` element displays as a `drop-down menu` (also called a pull-down menu) by default when `no size is specified` or if the `size` attribute is set to `1`.
- `Only one` item can be selected.

# Drop-down menus

<form>

<label for="pet-select">Choose a pet:</label>

<select id="pet-select" name="pet-select">

<option value="">--Please choose an option--</option>

<option value="dog">Dog</option>

<option value="cat">Cat</option>

<option value="hamster">Hamster</option>

<option value="parrot">Parrot</option>

<option value="spider">Spider</option>

<option value="goldfish">Goldfish</option>

</select>

</form>

Choose a pet: --Please choose an option--

Choose a pet ✓ --Please choose an option--

Dog

Cat

Hamster

Parrot

Spider

Goldfish

[Demo](#)

# Scrolling menus

- To make the menu display as a scrolling list
  - specify the number of lines to be visible using the size attribute (`size >1`).
- The `multiple` attribute allows users to make more than one selection from the scrolling list.
- When the browser detects the `multiple` attribute, it displays a small scrolling menu automatically by default.
- Use the `selected` attribute in an option element to make it the default value for the menu control.

# Scrolling menus

```
<form>
  <label for="pet-select">Choose a pet:</label>

  <select id="pet-select" name="pet-select" size="6">
    <option value="dog">Dog</option>
    <option value="cat">Cat</option>
    <option value="hamster">Hamster</option>
    <option value="parrot">Parrot</option>
    <option value="spider">Spider</option>
    <option value="goldfish">Goldfish</option>
  </select>
</form>
```

Choose a pet:

Dog  
Cat  
Hamster  
Parrot  
Spider  
Goldfish

[Demo](#)

# Scrolling menus with default value

```
<form>
```

```
<label for="pet-select">Choose a pet:</label>
```

```
<select id="pet-select" name="pet-select" size="6">
```

```
<option value="dog">Dog</option>
```

```
<option value="cat" selected>Cat</option>
```

```
<option value="hamster">Hamster</option>
```

```
<option value="parrot">Parrot</option>
```

```
<option value="spider">Spider</option>
```

```
<option value="goldfish">Goldfish</option>
```

```
</select>
```

```
</form>
```

Choose a pet:

Dog
Cat
Hamster
Parrot
Spider
Goldfish

# Scrolling menus with multiple

```
<form>
```

```
<label for="pet-select">Choose a pet:</label>
```

```
<select id="pet-select" name="pet-select" size="6" multiple>
```

```
<option value="dog">Dog</option>
```

```
<option value="cat">Cat</option>
```

```
<option value="hamster">Hamster</option>
```

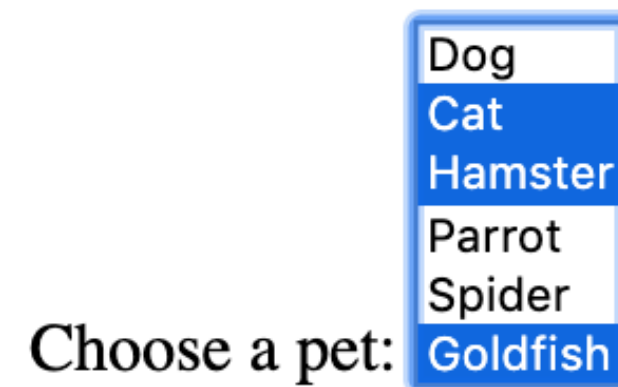
```
<option value="parrot">Parrot</option>
```

```
<option value="spider">Spider</option>
```

```
<option value="goldfish">Goldfish</option>
```

```
</select>
```

```
</form>
```



# Hidden controls

`<input type = "hidden" name="variable" value = "val">`

- The hidden control is used to send information to the form processing application that does not come from the user.
- The hidden control is not visible when the form is displayed in a browser. The hidden data is sent when the form is submitted.

`<input type = "hidden" name = "pagename" value = "10">`

# Submit and reset buttons

```
<input type = "submit">
```

Submits the form data to the server

```
<input type = "reset">
```

Resets the form controls to their default settings

- Both `submit` and `reset buttons` have specific functions that do not include the entry of the data.
- They are the only form control elements that `do not require` the `name` attribute.
- `By default`, the submit button displays with the label "`Submit`" or "`Submit Query`" and the reset button is labeled "`Reset`".
- Change the text on the button using the `value` attribute.

```
<input type = "submit" value = "Save">
```



# hidden controls

<form>

<p>This is page 10</p>

<input type = "hidden" name = "pagename" value = "10" />

<input type = "submit" name = "submit" value = "Submit" />

<input type = "reset" name = "reset" value = "Reset" />

</form>

This is page 10

Submit

Reset

[Demo](#)

# Fieldset and legend

- The <fieldset> element indicates a logical group of form controls.
- A <fieldset> may also include a <legend> element that provides a caption for the enclosed fields.

# <fieldset> element

<form>

<fieldset>

<legend>Choose your favorite monster</legend>

<input type="radio" id="kraken" name="monster">

<label for="kraken">Kraken</label><br/>

<input type="radio" id="sasquatch" name="monster">

<label for="sasquatch">Sasquatch</label><br/>

<input type="radio" id="mothman" name="monster">

<label for="mothman">Mothman</label>

</fieldset>

</form>

Choose your favorite monster

- ☐ Kraken
- ☐ Sasquatch
- ☐ Mothman

[Demo](#)

# References

- [https://www.tutorialspoint.com/html/html\\_forms.htm](https://www.tutorialspoint.com/html/html_forms.htm)
- [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)
- [https://www.w3schools.com/html/html\\_form\\_elements.asp](https://www.w3schools.com/html/html_form_elements.asp)