

WEB222 - Web Programming Principles

**Week 10: HTML Forms
& Form CSS Styling**

Agenda

- Introduction to Form
 - <input> tag
 - Text Fields in a Form
 - Selection Fields in a Form
 - Form Buttons
- Styling HTML Forms Using CSS
- More on CSS Selectors

HTML Forms

- HTML forms are web page components that are used to collect user input.
- An HTML form can contain **input elements / form controls**, such as:
 - input fields, text area, buttons, checkboxes, select lists, fieldset, legend, and label.
- HTML forms are originally used to pass data to a server.
 - The client fills out some information and then the browser sends the data from the form fields to the server for processing.
 - For front-end web application, HTML forms can also be used for **in-browser Processing** (without sending data to server).

HTML Forms

➤ Examples

□ Oracle User Registration

□ simple-form.html

The <form> Element

- The <form> element defines an HTML form:

```
<form id="formId" method="...."  
      action="url" >  
      .... ....  
</form>
```

- HTML forms contain **form elements** (or **form controls**).
- A document may have more than one form, but **forms cannot be nested**.

Attributes for <form> - method

➤ The **method** attribute

- It declares the **HTTP method** that the browser uses to send/post the form data to the server.
- **method="get"** :
 - ▶ is the default. The fill-out form contents to be appended to the URL as if they were a normal query (maximum of 256 characters).
 - ▶ Example: [simple-form.html](#)
- **method="post"** :
 - ▶ the fill-out form contents to be sent to the server in a data body rather than as part of the URL
 - ▶ Method "post" is more secure.
 - ▶ Example: [simple-form-post.html](#)

Attributes for <form> - action

- The **action** attribute tells browser where to post the form data when the form is submitted.

- **action="url"**

- Normally, a form is submitted to an URL on a web server.
 - Examples

```
<form id = "example1" method="post"  
      action="https://formpost.azurewebsites.net/home/test"> ... </form>
```

- The URLs for testing your forms:
 - <https://formpost.azurewebsites.net/home/test> (not for 'get', local)
 - <https://httpbin.org/get> (for 'get' method, JSON format)
 - <https://httpbin.org/post> (for 'post' method , JSON format)

- **action="#"**

- Submit to current page, used for in-browser Processing

Form Elements

- An HTML form can contain a number of form elements/fields/controls:
 - The **<input>** element is the one of the most-used form element.
 - Other form elements which are used gather or process user's input:
 - **<select>**, **<textarea>**, **<button>**.
 - Other elements that can be used in forms:
 - **<fieldset>**, **<legend>**, **<label>**, ...

The <input> Elements

- The **<input> tag** is used to specify a simple input element inside a form that can receive user input.
 - <input> tags are **inline, empty tags** . It may be written as <input />
- **type attribute.**
 - The **type** indicates what sort of input field the tag represents, such as text boxes or radio buttons.
 - By default, type="text".
- **name attribute**
 - All <input> tags are **required** to have **name attribute and value**, except for the submit and reset buttons.
 - Without name attribute, the user's input for this input element will not be sent to server.

Type Attributes of the <input> Tag

➤ type="text"

- A text element is a single line text input field in which the user can enter text.
- type="text" is **default** for the tag.
- Other attributes for this type: **size**, **maxlength**

Text field 1 <input type="text" name="entry1" id="entry1" />text box default size = 20

Text field 2 <input size="30" maxlength="10" name="entry2" id="entry2"/>

Text field 3 <input size="5" maxlength="10" name="entry3" id="entry3">

Text field 4 <input size="12" value="416-" name="entry4" id="entry4" >

Text field 1	<input type="text"/>	text box default size = 20
Text field 2	<input type="text"/>	
Text field 3	<input type="text"/>	
Text field 4	<input type="text" value="416-"/>	

❑ [input-tags-1.html](#)

Type Attributes of the <input> Tag

➤ type="password"

- A password element is a text input field in which each character typed is displayed as a character such as * or a black dot to conceal the actual value.

Type in your username <input name="username" />

Type in your password <input type="password" name="password" />

Type in your username

Type in your password

Type Attributes of the <input> Tag

➤ type="hidden"

- A hidden input element is an invisible element whose main purpose is to contain data that the user does not enter. This data gets sent to the invoked server program when the form is submitted.
- The type="hidden" attribute provides a way for delivering a value to server program

```
<input type="hidden" name="entry0" id="entry0" value="value from the  
form" />
```

Type Attributes of the <input> Tag

➤ type="file"

- A file element allows the user to supply a file as input. When the form is submitted, the content of the specified file is sent to the server as the value portion of the name/value pair for this input element.
- A 'Browse' button is displayed next to the file input element that lets users select a file from their system to use as the value of the file input element.
- If a form contains a file input element, the value of the enctype attribute of the form tag should be 'multipart/form-data'.

```
Student Name <input type="text" name="StudentName" id="StudentName"><br>
Upload your assignment <input type="file" name="assignment" multiple>
```

Student Name

Upload your assignment

Browse...

No file selected.

Type Attributes of the <input> Tag

➤ type="button"

```
<input type="button" value="Press This Button" />
```

➤ type="image"

- Places an image, serving as a custom **button** in place of the submit button. When a user clicks the image, the form is submitted to the server.

Search

Search

Go Get It

[input-tags-1.html](#)

[gogetit.gif](#)

Type Attributes of the <input> Tag

- type="submit"
 - When a user clicks a submit button, the form is submitted to the "url" specified by the action attribute of the form.
 - If no value attribute: "Submit Query" will show on the button.
- type="reset"
 - When a user clicks a reset button, all elements in the form are reset to their original
 - Default value for value attribute: "Reset".

```
<p>
  <input type="submit" value='Submit' >
  <input type="reset" value=" Clear ">
</p>
```

Submit

Clear

Type Attributes of the <input> Tag

➤ type="checkbox"

- A checkbox element is a toggle that the user can select (switch on) or deselect (switch off.)
- All checkbox items **should have the same name** indicating they are in the same group.

<p>Which operating system do you use? </p>

```
<input type="checkbox" name="system_type" id="stype-2" value="2" />Windows 7<br />
<input type="checkbox" name="system_type" id="stype-3" value="3" checked>Windows 8<br />
<input type="checkbox" name="system_type" id="stype-4" value="4" checked/> Unix<br />
```

Which operating system do you use?

- Windows 7
- Windows 8
- Unix

➤ Notes: 1. <input> tag is an **empty tag**. 2. the **name** and **value** must be specified so that the name and value pair can be sent to server.

□ [input-tags-2.html](#)

Type Attributes of the <input> Tag

➤ type="radio"

- A radio element is a radio button.
- Only one radio button in the set can be selected at one time..
- All radio button items **must have the same name** indicating they are in the same group.

```
<ul>
<li><input type="radio" name="paymethod" id="paymethod-1" value="cash" checked />Cash</li>
<li><input type="radio" name="paymethod" id="paymethod-2" value="cheque" />Cheque</li>
<li><mark>Credit card</mark>
<ul>
  <li><input type="radio" name="paymethod" id="paymethod-3" value="mastercard" />
Mastercard</li>
  <li><input type="radio" name="paymethod" id="paymethod-4" value="visa" />Visa</li>
</ul></li>
</ul>
```

- Cash
- Cheque
- Credit card
 - Mastercard
 - Visa

Text field with <datalist> tag

- HTML5 <**datalist**> tag: specifies a list of pre-defined options for an <input type="text"> element.
- To provide an "autocomplete" feature on <input> elements.
- Users will see a drop-down list of pre-defined options as they input data.
- e.g.

```
<input type="text" list="subjects" name="course" />
<datalist id="subjects">
    <option value="EAC150">
    <option value="IPC144">
    <option value="ULI101">
    <option value="IOS110">
</datalist>
```

Additional Attributes for <input> Tag

- The following are additional (other than type) attributes that work with the input tag.
 - Global attributes: id, class, style, title, tabindex
 - <input> specific attributes: name, value, checked, size, maxlength, disabled, readonly.
 - The **name & value** pair in each input tag is what is going to be send to server.
 - Some or all of these attributes may be used depending on the value of the type attribute.

➤ e.g.

□ [input-tags-attributes.html](#)

Attributes of the <input> Tag

➤ **name**

- The name attribute and a value should be present for all input tags. Otherwise, it cannot be sent to the server.

➤ **Value**

- for a text or password entry field, can be used to specify the default contents of the field.
 - ▶ **default value** or **current value**
- for each checkbox or radio button, value has to be specified. Otherwise no value of the button will be sent out even it is checked. Unchecked buttons are ignored when submitting the form.
- for types submit and reset buttons, value is used to specify the text on the buttons.

Additional Attributes of the <input> Tag

➤ **checked**

- checked="checked" specifies that the checkbox or radio button is checked by default;
- HTML5 supports attribute minimization – use **checked** for simplifying *checked="checked"*.

➤ **size**

- is the physical size of the input field in characters;
- this is appropriate for text entry fields and password entry fields.
- If this is not present, the **default is 20** characters.

Additional Attributes of the <input> Tag

➤ **maxlength**

- is the maximum number of characters that are accepted as input; this is only appropriate for single-line text entry fields and password entry fields.
- If this is not present, the default will be **unlimited**.
- The text entry field will scroll appropriately if maxlength value is greater than the size value.

➤ **tabindex**

- tabindex="nn" - nn is a positive value - navigation proceeds from the element with the lowest tabindex value to the element with the highest value.
- tabindex is a **global attribute**.

❑ tabindex.html

The diagram illustrates a form layout with four text input fields and two buttons. Orange arrows indicate the tab index flow: it starts at the bottom left button, moves right to the first text input field labeled 'Text 1', then up to the second text input field labeled 'Text 2'. From 'Text 2', the flow goes down to the third text input field labeled 'Text 3', then right to the fourth text input field labeled 'Text 4'. Finally, the flow goes back down to the bottom right button labeled 'Clear'.

Text 1		Text 2
Text 3		Text 4
Submit Information		Clear

Additional Attributes of the <input> Tag

➤ **disabled**

- `disabled="disabled"` - When used, cannot receive user input nor will its value be submitted to server with the form

➤ **readonly**

- `readonly="readonly"` - When used, cannot receive user input - the value is submitted to server with the form

➤ **autocomplete** attribute

- Specifies whether a HTML form or its form elements have autocomplete on or off.

[autocomplete.html](#)

HTML5 Form input Types & Attributes



- HTML5 introduces many semantic input types to enrich the “text” like type.
 - The new input types include: email, url, tel, number, range, date, month, week, time, datatime, color, search
 - e.g.
`<input type="number" name="even" min="2" max="100" step="2" />`
 - No matter what type it is, the value is always "text".
- HTML5 also supports some new attributes for form input elements.
 - autofocus, placeholder, required

HTML5 Form input Types & Attribute



The FORM tag - filling-out form example using new html5 features

html5 Focus on an entry **autofocus**

First name goes here **placeholder**

required

mm / dd / yyyy **type="date"**

----- , ----- **type="month"**

-- : -- -- **type="time"**

----- **type="datetime"**

Week -- , ----- **type="week"**

type="color"

type="number" **with min & max** **with min, max & step**

Subjects **Click here to select** **type="text" with a list**

type="email"

type="url"

type="phone"

type="search"

[input-tags-html5.html](#)

More Form Elements

- In addition to `<input>` element, some elements can be used inside a form to accept user input:
 - `<select>` element
 - with `<option>` and `<optgroup>` elements
 - `<textarea>` element
 - `<button>` element

<select> Element

- The <select> element is used to create a **drop-down list**, from where a user can select one or more options.
- E.g.

```
<select name="what-to-do" id="what-to-do" size=5>
  <option value="1"> Drink Coffee </option>
  <option value="2" selected> Read A Book </option>
  <option value="3"> Take A Walk </option>
  <option value="4"> Buy A Bagel </option>
  <option value="5"> Watch TV </option>
  <option value="6"> Write a test </option>
</select>
```

<select> Element

- The selection list itself is defined by a series of <option> tags.
 - The **name** attribute is in the <select> tag; while the **value** attribute is in all option tags.
 - If more than one option is selected in the list, the selected options are all sent to the server as a comma separated list under that one name.
- [select-tags-attributes.html](#)

The attributes of the <select> tag

➤ **multiple="multiple"**

- allows users to select more than one option, usually by holding down the Control key while clicking on additional choices.
- Otherwise the selection functions like radio buttons where selecting one deselects another.

➤ **size="n"**

- specify how many lines are displayed in the selection menu.
- If the size is not specified or if size="1", a single line is displayed and the selection menu functions as a drop down menu.
- If a number larger than one is specified, then the menu functions as a scrollable list.

<option> Element

- The <option> tag is the one in a form which has “text”
- value vs text

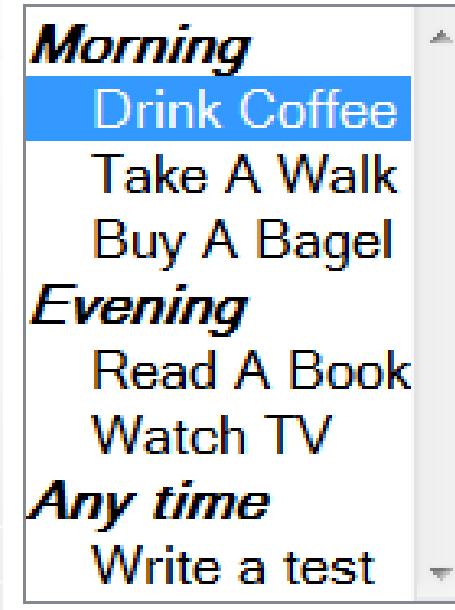
e.g. <select name="thename">
 <option value="1">Text 1</option>
 <option value="2">Text 2</option>
 </select>

- The value attribute is the value that is returned by selecting that option.
- If a <option> tag's value attribute is not provided, its text will be come the value.

<select> Element with <optgroup>

- The <optgroup> tag to group things by category.

```
<select name="what-to-do" id="what-to-do"
        multiple="multiple" size="9">
    <optgroup label="Morning">
        <option value="1"
               selected="selected"> Drink Coffee
        </option>
        <option value="2"> Take A Walk </option>
        <option value="3"> Buy A Bagel </option>
    </optgroup>
    <optgroup label="Evening">
        <option value="4"> Read A Book </option>
        <option value="5"> Watch TV </option>
    </optgroup>
    <optgroup label="Any time">
        <option value="6"> Write a test </option>
    </optgroup>
</select>
```



□ [select-tags-optgroup.html](#)

<textarea> Element

- The <textarea> element provides a multi-line text entry field.
- attributes:
 - rows = "height of the textarea in character"
 - cols = "width of the textarea in character"
- textarea fields automatically have scroll bars; any amount of text can be entered in them.
- The element can have "text" as default contents. E.g.

```
<textarea name="comments" cols="30" rows="10">  
Input your comments:  
</textarea>
```

textarea.html

<button> Element

- The <button> element provide a alternative way to perform most of the above actions with greater flexibility.
- Syntax:
 - <button> content showing on button </button>
- Type attributes:
 - <button type="submit">
 - <button type="reset">
 - <button type="button">
- Example 1 - A button located outside a form (but still a part of the form):

```
<button type="submit" form="form1" value="Submit">Submit</button>
```
- Example 2 – A button doesn't belong to any form.

```
<button type="button" onclick="alert('Button was clicked!')>  
Click Me!</button>
```

Grouping Fields

- <**fieldset**> tags: grouping the fields
- <**legend**> tags: specifying a title for the group

```
<fieldset>
```

```
  <legend>Personal Information</legend>
```

```
    Frist Name: <input type="text" name="fname" id="fanme" /><br>
```

```
    Last Name: <input type="text" name="lname" id="lname" /> <br>
```

```
    Email: <input type="email" name="name3" id="email" /> <br />
```

```
    Telephone: <input type="tel" name="phone" id="phone" value="416-"/>
```

```
</fieldset>
```

❑ [fieldset-label-button.html](#)

<label> Element

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

<label> Element

➤ Example

```
<h4>Label exempel 1</h4>
<div>
  <label for="entry1" title="Free format">Text field 1</label>
  <input type="text" name="entry1" id="entry1" />
</div>
```

```
<h4>Label exempel 2</h4>
<p>
  <label for="entry3" title="Label for entry3">
    <span>Text field 3 </span>
    <input size="5" name="entry3" id="entry3" /> <br />
  </label>
</p>
```

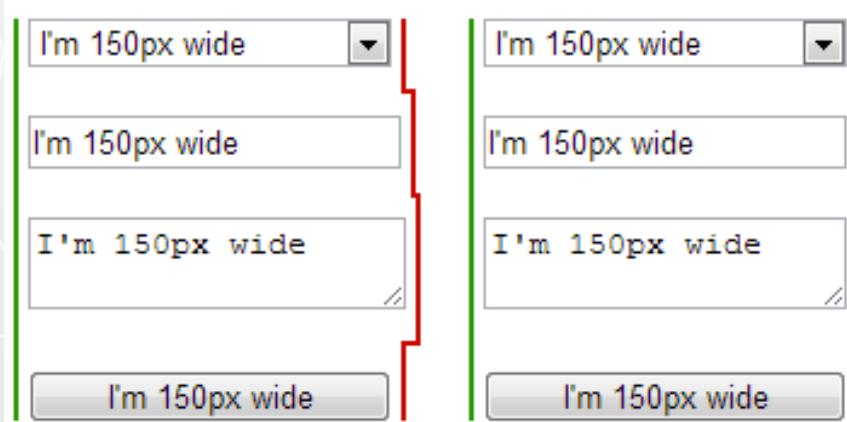
□ [fieldset-label-button.html](#)

Styling HTML Forms Using CSS

Issues:

- HTML table is not appropriate for aligning form elements.
- Not all form elements are created equal when CSS is involved.
 - Some elements are easy to style with CSS, but some other elements simply can't be styled using CSS.
- Browsers are often inconsistent in rendering font and text.

```
button, input, select,  
textarea  
{  
    font-family : inherit;  
    font-size : 100%;  
    width : 150px;  
}
```



Styling HTML Forms Using CSS

➤ CSS box-sizing property:

"box-sizing: border-box;" can be used to give the same size to several different form elements.

- Example

```
input, textarea, select, button {  
    width : 150px;  
    margin: 0;  
    box-sizing: border-box;  
}
```

➤ CSS property "display:inline-block; "

- <label> and are inline-level elements, so width and height are available to them.
- "display:inline-block;" can be used to inline-level elements in order to set dimensions (width and height) to them.
- Example:

```
label { display: inline-block; width: 100px; text-align: right; }
```

An example form with CSS

```
<form action="/my-handling-form-page" method="post">  
  <div>  
    <label for="name">Name:</label>  
    <input type="text" id="name" />  
  </div>  
  <div>  
    <label for="mail">E-mail:</label>  
    <input type="email" id="mail" />  
  </div>  
  <div>  
    <label for="msg">Message:</label>  
    <textarea id="msg"></textarea>  
  </div>  
</form>
```

HTML Form with CSS

Name:

E-mail:

Message:

Send your message

An example form with CSS

- Example: [form-with-css.html](#)

After using CSS:

HTML Form with CSS

Name:

E-mail:

Message:

- Here is an other example from MDN: [A payment form](#)

More on CSS

➤ Attribute selector

- An attribute selector will match elements on the basis of either
 - the presence of an attribute, or
 - the exact or partial match of an attribute value.
- e.g.

```
input[type="submit"] { border: 2px solid #ccc; }  
[class="warning"] { background-color: yellow; }
```

□ morecss.html

HTML Form Summary

➤ Form element:

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

➤ Form elements/controls:

- <**input**> type =:
 - ▶ Textboxes: **text**, **password**, **email**, **file**, **color**, **date**, **time**,
number, **url**, **tel**, ...
 - ▶ Selections: **checkbox**, **radio**
 - ▶ Buttons: **submit**, **reset**, **image**, **button**
- Other elements:
 - <**select**> with <**option**>, <**textarea**>, <**button**>.
 - <**fieldset**>, <**legend**>, <**label**>, ...

Advanced: More on CSS Selectors

(Notes: for “Advanced” part, read only if you’re interested in it)

➤ Type of Selectors

- Type selector (tag selector)
- Class selector
- ID selector
- Selector grouping
- Universal selector
- Attribute selector
- Combinators
 - Descendant selector (contextual selectors)
 - Child selector
 - Adjacent sibling selector
 - General sibling selector
- Pseudo-classes
- Pseudo-elements

Advanced: Selectors

➤ Universal selector

- The universal selector matches any element type.
- e.g.
`* { margin: 0; padding: 0; }`

➤ Attribute selector

- An attribute selector will match elements on the basis of either
 - the presence of an attribute, or
 - the exact or partial match of an attribute value.
- e.g.

`[href] { color: yellow; }`

`a[href] {color: yellow;}`

`input[type="submit"] { border: 2px solid #ccc; }`

`[class="warning"] { background-color: yellow; }`

Advanced: Selectors

➤ Combinators - Selectors based on relationships

- Descendant selector (contextual selectors)
 - e.g. `div span` { line-height: 90%; } // includes child
- Child selector
 - e.g. `ul>li` { text-decoration: underline; }
- Adjacent sibling selector
 - e.g. `h2+p` { color: blue; } // matches all p elements that appear immediately after h2 elements.
- General sibling selector
 - e.g. `h2~p` { color: blue; }

Advanced: Selectors

➤ Pseudo-class selector

- A pseudo-class is similar to a class in HTML, but it's not specified explicitly in the markup.

:active, :hover, :link, :visited, :focus, :first-child, ...

➤ Pseudo-element selector

- allow you to style certain parts of a document:

- CSS2

:after, :before, :first-letter, :first-line, :selection

- CSS3

::after, ::before, ::first-letter, ::first-line, ::selection

Resourceful Links

- [W3C HTML5 DOC: Forms](#)
- [MDN: Forms in HTML](#)
- [How to structure an HTML form](#)
- [Selectors - Web developer guide | MDN](#)
- [**Web Dep Tool: Firefox Style Editor**](#)

Thank You!