



University of Tripoli
Faculty of Information Technology



Department of Software Engineering

HTML5 BASICS (4)

Introduction to Internet Programming
ITGS 226 -- F 2023

By: Fatima Ben Lashihar

HTML FORMS

- HTML Forms are required to collect different kinds of user inputs, such as contact details like name, email address, phone numbers, or details like credit card information, etc.
- The `<form>` element is a container for different types of input elements, such as: text fields, checkboxes, radio buttons, submit buttons, etc... called **controls**.
- Users generally complete a form by modifying its controls e.g. entering text, selecting items, etc. and submitting this form to a web server for further processing.

HTML FORMS (cont'd) **Form Attributes**

- The **action** attribute:

Specifies the URL of the program or script on the web server that will be used for processing the information submitted via form. If the action attribute is omitted, the action is set to the current page.
- The **method** attribute:

Specifies the HTTP method used for sending the data to the web server by the browser. The value can be either get (the default) and post . All the data sent via get method is visible in the browser's address bar. But, the data sent via post is not visible to the user. **Ex**
- The **name** attribute:

Specifies the name of the form. Its value must be unique among the forms in a document, and must not be an empty string.

HTML FORMS (cont'd) **Form Attributes**

- The **target** attribute:

Specifies where to display the response that is received after submitting the form. Possible values are `_blank` , `_self` , `_parent` and `_top`. The default value is `_self` which means that the response will open in the current window. [Example](#)
- The **autocomplete** attribute:

Specifies whether a form should have autocomplete on or off. When autocomplete is on, the browser automatically complete values based on values that the user has entered before. The default value is on. [Example](#)
- The **novalidate** attribute:

Specifies that the form-data (input) should not be validated when submitted. [Example](#)

HTML FORMS (cont'd) **Form elements**

- <fieldset> & <legend>
- <label>
- <input> & its types & its attributes
- <textarea>
- <select> & <option> & <optgroup>
- <datalist>
- <button>

HTML FORMS (cont'd) **<fieldset>**

- The `<fieldset>` element is used to group related data in a form.
- The `<fieldset>` tag draws a box around the related elements.
- The `<legend>` element defines a caption for the `<fieldset>` element.

Example

HTML FORMS (cont'd) **<label>**

- The `<label>` tag defines a label for several elements such as `input`, `textarea`, `select`, etc.
- labels will benefit screen reader users (will read out loud the label, when the user is focused on the element) and users who have difficulty clicking on very small regions (such as checkboxes) - because when a user clicks the text within the `<label>` element, it toggles the input (this increases the hit area).
- The `for` attribute of `<label>` must be equal to the `id` attribute of the related element to bind them together.
- A label can also be bound to an element by placing the element inside the `<label>` element.

HTML FORMS (cont'd) **<Input>**

- This is the most commonly used element within HTML forms.
- An input element can be of type *text field*, *password field*, *checkbox*, *radio button*, *submit button*, *reset button*, *file select box*, as well as several new input types introduced in HTML5.
- The default value of the type attribute is "text".

HTML FORMS (cont'd) **<Input type ="">**

- `<input type="text">`
- `<input type="password">`
- `<input type="hidden">`
- `<input type="submit">`
- `<input type="reset">`
- `<input type="button">`
- `<input type="image">`
- `<input type="radio">`
- `<input type="checkbox">`
- `<input type="file">`
- `<input type="date">`
- `<input type="datetime-local">`
- `<input type="time">`
- `<input type="week">`
- `<input type="month">`
- `<input type="search">`
- `<input type="tel">`
- `<input type="url">`
- `<input type="email">`
- `<input type="number">`
- `<input type="range">`
- `<input type="color">`

HTML FORMS (cont'd) `<input type="">`

- `<input type="text">`
defines a single-line text input field. [Example](#)
- `<input type="password">`
defines a password field. The characters in a password field are masked (shown as asterisks or circles). [Example](#)
- The `<input type="hidden">`
defines a hidden input field (not visible to a user). This field let web developers include data that cannot be seen or modified by users when a form is submitted. [Example](#)

HTML FORMS (cont'd) `<input type="">`

- `<input type="submit">`

defines a button for submitting form data to a form-handler which is typically a server page with a script for processing input data that is specified in the form's action attribute. [Example](#)

- `<input type="reset">`

defines a reset button that will reset all form values to their default values. [Example](#)

- `<input type="button">`

defines a button, mostly used with a JavaScript to activate a script Inside a `<button>`. [Example](#)

- `<input type="image">`

defines an image as a submit button. The path to the image is specified in the src attribute. [Example](#)

HTML FORMS (cont'd) `<Input type ="">`

- `<input type="radio">`

defines a radio button. Radio buttons let a user select ONLY ONE of a limited number of choices. The radio group must have share the same name (the value of the name attribute) to be treated as a group. To make a radio button selected by default, add the attribute `checked` to the input element. [Example](#)

- `<input type="checkbox">`

defines a checkbox. Checkboxes let a user select ZERO or MORE options of a limited number of choices. The name is used to group multiple checkboxes together. To make a checkbox selected by default, add the attribute `checked` to the input element. [Example](#)

- `<input type="file">`

defines a file-select field and a "Browse" button for file uploads. It allows a user to browse for a local file and send it as an attachment with the form data. It has the attribute `accept` that specifies a filter for what file types the user can pick from the file input dialog box e.g. `<input accept="audio/*,video/*,image/*" >` or `<input accept=".jpg">`. [Example](#)

HTML FORMS (cont'd) `<input type="">`

- `<input type="date">`
is used for input fields that should contain a date. The resulting value includes the year, month and day.
- `<input type="datetime-local">`
specifies a date and time input field, with no time zone. The resulting value includes the year, month, day and time.
- `<input type="time">`
allows the user to select a time (no time zone).
- `<input type="week">`
allows the user to select a week and year.
- `<input type="month">`
allows the user to select a month and year. The format is "YYYY-MM".

Example

HTML FORMS (cont'd) `<input type="">`

- The `<input type="search">`
is used for search fields (a search field behaves like a regular text field).
- The `<input type="tel">`
is used for input fields that should contain a telephone number.
- The `<input type="url">`
is used for input fields that should contain a URL address.
- `<input type="email">`
is used for input fields that should contain an e-mail address. The input value is automatically validated to ensure it is a properly formatted e-mail address.

Example

HTML FORMS (cont'd) `<input type="">`

- `<input type="number">`
defines a numeric input field. Restrictions can be set using the attributes: `max`, `min`, `step` or `value`.
- The `<input type="range">`
defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the `min`, `max`, and `step` attributes
- `<input type="color">`
used for input fields that should contain a color. The default value is #000000 (black).

Example

HTML FORMS (cont'd) **<Input> attributes**

Attribute	Value	Description
name	text	Specifies the name of an <input> element
required	required	Specifies that an input field must be filled out before submitting the form
autofocus	autofocus	Specifies that an <input> element should automatically get focus when the page loads
readonly	readonly	Specifies that an input field is read-only
disabled	disabled	Specifies that an <input> element should be disabled
placeholder	text	Specifies a short hint that describes the expected value of an <input> element
autocomplete	on off	Specifies whether an <input> element should have autocomplete enabled
Accept	file_extension audio/* video/* image/*	Specifies a filter for what file types the user can pick from the file input dialog box (only for type="file")
checked	checked	Specifies that an <input> element should be pre-selected when the page loads (for type="checkbox" or type="radio")
max	number date	Specifies the maximum value for an <input> element
maxlength	number	Specifies the maximum number of characters allowed in an <input> element
min	number date	Specifies a minimum value for an <input> element
minlength	number	Specifies the minimum number of characters required in an <input> element
multiple	multiple	Specifies that a user can enter more than one value in an <input> element
pattern	regexp	Specifies a regular expression that an <input> element's value is checked against
size	number	Specifies the width, in characters, of an <input> element
step	number	Specifies the interval between legal numbers in an input field
value	text	Specifies the value of an <input> element

Example

HTML FORMS (cont'd) `<textarea>`

- `<textarea>` is a multiple-line text input control that allows a user to enter more than one line of text.
- The size of a text area is specified by the `<cols>` and `<rows>` attributes (or with CSS).

```
<form>
  <label for="address">Address:</label>
  <textarea rows="3"
            cols="30"
            name="address"
            id="address">
  </textarea>
</form>
```

HTML FORMS (cont'd) `<textarea>` attributes

Attribute	Value	Description
cols	number	Specifies the visible width of a text area
rows	number	Specifies the visible number of lines in a text area
wrap	hard soft	Specifies how the text in a text area is to be wrapped when submitted in a form EX
name	text	Specifies a name for a text area
required	required	Specifies that a text area is required/must be filled out
autofocus	autofocus	Specifies that a text area should automatically get focus when the page loads
readonly	readonly	Specifies that a text area should be read-only
disabled	disabled	Specifies that a text area should be disabled
placeholder	text	Specifies a short hint that describes the expected value of a text area
maxlength	number	Specifies the maximum number of characters allowed in the text area

HTML FORMS (cont'd) `<select>`

- **Select Boxes** is a dropdown list of options that allows user to select one or more option from a pull down list of options.
- Select box is created using the `<select>` element and `<option>` element.
- The `<option>` elements within the `<select>` element define each list item.
- By default, the first item in the drop-down list is selected.
- To define a pre-selected option, add the `selected` attribute to the option.
- Use the `size` attribute to specify the number of visible values:
- Use the `multiple` attribute to allow the user to select more than one value.
- The `<optgroup>` tag is used to group related options in a `<select>` element (drop-down list).

Example

HTML FORMS (cont'd) `<select>` attributes

Attribute	Value	Description
name	name	Defines a name for the drop-down list
required	required	Specifies that the user is required to select a value before submitting the form
autofocus	autofocus	Specifies that the drop-down list should automatically get focus when the page loads
disabled	disabled	Specifies that a drop-down list should be disabled
size	number	Defines the number of visible options in a drop-down list
multiple	multiple	Specifies that multiple options can be selected at once

HTML FORMS (cont'd) `<datalist>`

- The `<datalist>` tag specifies a list of pre-defined options for an `<input>` element.
- The `<datalist>` tag is used to provide an "autocomplete" feature for `<input>` elements. Users will see a drop-down list of pre-defined options as they input data.

Example

HTML FORMS (cont'd)

Differences between select and datalist

- Select is an form input type while Datalist isn't.
- Select input element presents options for the users from which they need to select one of them. On the other hand, Datalist presents a list of suggested values to the associated input form (text) field and users are free to select one of those suggested values or type in their own value.
- With Select, users may have to scan a long list for selecting one of the values, while with Datalist, the values are provided as hints and users aren't bound to those values. They could rather provide their own inputs.

Example

HTML FORMS (cont'd) `<button>`

- The `<button>` tag defines a clickable button.
- Inside a `<button>` element you can put text (and tags like `<i>`, ``, ``, `
`, ``, etc.). That is not possible with a button created with the `<input>` element!
- Always specify the `type` attribute for a `<button>` element, to tell browsers what type of button it is: `button`, `submit` or `reset`.

Example

REGULAR EXPRESSION

- Regular Expressions, commonly known as "**regex**" or "**RegExp**", are a specially formatted text strings used to find patterns in text.
- It can be used to verify whether the format of data i.e. name, email, phone number, etc. entered by the user was correct or not, find or replace matching string within text content, and so on.
- Regular expression syntax includes the use of special characters. The characters that are given special meaning within a regular expression, are: `. * ? + [] () { } ^ $ | \`. You will need to backslash these characters whenever you want to use them literally.

REGULAR EXPRESSION

Character Classes

RegExp	What it Does
[abc]	Matches any one of the characters a, b, or c.
[^abc]	Matches any one character other than a, b, or c.
[a-z]	Matches any one character from lowercase a to lowercase z.
[A-Z]	Matches any one character from uppercase a to uppercase z.
[a-Z]	Matches any one character from lowercase a to uppercase Z.
[0-9]	Matches a single digit between 0 and 9.
[a-z0-9]	Matches a single character between a and z or between 0 and 9.

Example

REGULAR EXPRESSION

Repetition Quantifiers

RegExp	What it Does
p+	Matches one or more occurrences of the letter p.
p*	Matches zero or more occurrences of the letter p.
p?	Matches zero or one occurrences of the letter p.
p{2}	Matches exactly two occurrences of the letter p.
p{2,3}	Matches at least two occurrences of the letter p, but not more than three occurrences of the letter p.
p{2,}	Matches two or more occurrences of the letter p.
p{,3}	Matches at most three occurrences of the letter p

Example

THE END