



HTML FORMS

**SUBJECT: BASICS OF WEB
DESIGN (09CE2102)**

DEPTRTMENT: CE-DIPLO

BY: ZEEL MEHTA

CONTENT

- HTML Forms
- HTML Form Attributes **(FOR REFERENCE ONLY)**
- HTML Form Elements
- HTML Input Types
- HTML Input Attributes

HTML FORM

An HTML form is a section of a document which contains controls such as text fields, password fields, checkboxes, radio buttons, submit button, menus etc.

An HTML form facilitates the user to enter data that is to be sent to the server for processing such as name, email address, password, phone number, etc. .

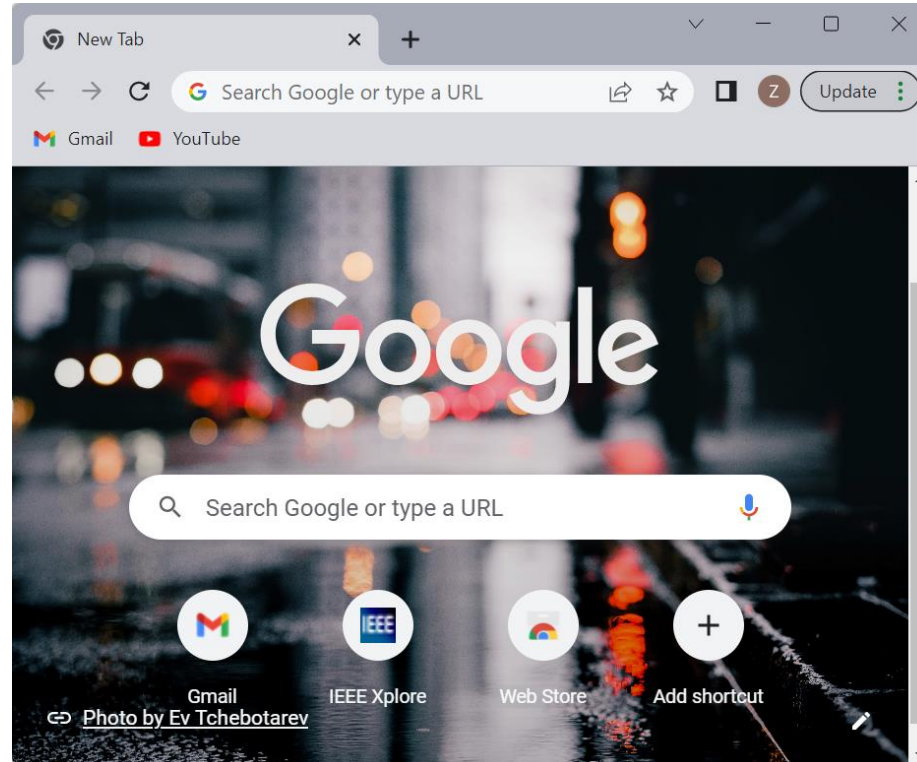
An HTML form is used to collect user input. The user input is most often sent to a server for processing.

HTML forms are required if you want to collect some data from of the site visitor.

For example: If a user want to purchase some items on internet, he/she must fill the form such as shipping address and credit/debit card details so that item can be sent to the given address.

CONTINUE..

The best known form on the web is probably the search box that sits right in the middle of Google's homepage.



HTML Form Attributes

This topic describes the different attributes for the HTML <form> element.

NOTE: THIS IS ONLY FOR YOUR REFERENCE AND KNOWLEDGE.

The Action Attribute

- The **action** attribute defines the action to be performed when the form is submitted.
- Usually, the form data is sent to a file on the server when the user clicks on the submit button.
- In the example below, the form data is sent to a file called "action.php". This file contains a server-side script that handles the form data from "form.html".

CONTINUE..

The Target Attribute

- The target attribute specifies where to display the response that is received after submitting the form.
- **EXAMPLE:** Here, the submitted result will open in a new browser tab:
`<form action="/action_page.php" target="_blank">`
- The target attribute can have one of the following values:

**NOTE: THIS IS ONLY FOR
YOUR REFERENCE
AND KNOWLEDGE.**

Value	Description
_blank	The response is displayed in a new window or tab
_self	The response is displayed in the current window
_parent	The response is displayed in the parent frame
_top	The response is displayed in the full body of the window
framename	The response is displayed in a named iframe

The Autocomplete Attribute

- 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.

➤ Example:

A form with autocomplete on:

```
<form action="/action_page.php" autocomplete="on">
```

NOTE: THIS IS ONLY FOR YOUR REFERENCE AND KNOWLEDGE.

The Novalidate Attribute

- The **novalidate** attribute is a boolean attribute.
- When present, it specifies that the form-data (input) should not be validated when submitted.

- **Example:**

A form with a novalidate attribute:

```
<form action="/action_page.php" novalidate>
```

NOTE: THIS IS ONLY FOR YOUR REFERENCE AND KNOWLEDGE.

CONTINUE..

The Method Attribute

- The **method** attribute specifies the HTTP method to be used when submitting the form data.
- The form-data can be sent as URL variables (with **method="get"**) or as HTTP post transaction (with **method="post"**).
- The default HTTP method when submitting form data is GET.
- Example:

This example uses the GET method when submitting the form data:

```
<form action="/action_page.php" method="get">
```

OR

```
<form action="/action_page.php" method="post">
```

Notes on GET:

- Appends the form data to the URL, in name/value pairs
- NEVER use GET to send sensitive data! (the submitted form data is visible in the URL!)
- The length of a URL is limited (2048 characters)
- Useful for form submissions where a user wants to bookmark the result
- GET is good for non-secure data, like query strings in Google

Notes on POST:

- Appends the form data inside the body of the HTTP request (the submitted form data is not shown in the URL)
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

Tip: Always use POST if the form data contains sensitive or personal information!

HTML Form Elements

The HTML `<form>` element can contain one or more of the following form elements:

- `<input>`
- `<label>`
- `<select>`
- `<textarea>`
- `<button>`
- `<fieldset>`
- `<legend>`
- `<datalist>`
- `<output>`
- `<option>`
- `<optgroup>`

Form tags and description

Tag	Description
<u><form></u>	Defines an HTML form for user input
<u><input></u>	Defines an input control
<u><textarea></u>	Defines a multiline input control (text area)
<u><label></u>	Defines a label for an <input> element
<u><fieldset></u>	Groups related elements in a form
<u><legend></u>	Defines a caption for a <fieldset> element
<u><select></u>	Defines a drop-down list
<u><optgroup></u>	Defines a group of related options in a drop-down list
<u><option></u>	Defines an option in a drop-down list
<u><button></u>	Defines a clickable button
<u><datalist></u>	Specifies a list of pre-defined options for input controls
<u><output></u>	Defines the result of a calculation

<form>

➤ The HTML <form> element provide a document section to take input from user. It provides various interactive controls for submitting information to web server such as text field, text area, password field, etc.

➤ Syntax:

```
<form>
```

```
//Form elements
```

```
</form>
```

Note: The <form> element does not itself create a form but it is container to contain all required form elements, such as <input>, <label>, etc.

<label>

- The <label> element defines a label for several form elements.
- The <label> element is useful for screen-reader users, because the screen-reader will read out loud the label when the user focus on the input element.
- The <label> element also help users who have difficulty clicking on very small regions (such as radio buttons or checkboxes) - because when the user clicks the text within the <label> element, it toggles the radio button/checkbox.
- The for attribute of the <label> tag should be equal to the id attribute of the <input> element to bind them together.

➤ Syntax:

```
<form>
```

```
  <label for="firstname">First Name: </label>
```

```
    <input type="text" id="firstname" name="firstname"> <br/>
```

```
  <label for>Last Name: </label>
```

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

```
</form>
```

NOTE: It is good to use <label> tag with form, although it is optional but if you will use it, then it will provide a focus when you tap or click on label tag. It is more worthy with touchscreens.

<input>

- The <input> tag specifies an input field where the user can enter data.
- The <input> element is the most important form element.
- The <input> element can be displayed in several ways, depending on the type attribute.

➤ Syntax:

```
<form>
```

```
  <label>First name:</label>
```

```
  <input type="text" id="fname" name="fname"><br>
```

```
  <label>password:</label>
```

```
  <input type="password" id="lname" name="lname"><br>
```

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

```
</form>
```

Id attribute: the id attribute **specifies a unique id for an HTML element**. The value of the id attribute must be unique within the HTML document. The id attribute is used to point to a specific style declaration in a style sheet. It is also used by JavaScript to access and manipulate the element with the specific id.

Name attribute: the name attribute **specifies the name of an <input> element**. The name attribute is used to reference elements in a JavaScript, or to reference form data after a form is submitted. Note: Only form elements with a name attribute will have their values passed when submitting a form.

Value attribute: the value attribute **specifies the value of an <input> element**. The value attribute is used differently for different input types: For "button", "reset", and "submit" - it defines the text on the button. For "text", "password", and "hidden" - it defines the initial (default) value of the input field.

<textarea>

- The <textarea> tag defines a multi-line text input control.
- The <textarea> element is often used in a form, to collect user inputs like comments or reviews.
- A text area can hold an unlimited number of characters, and the text renders in a fixed-width font (usually Courier).
- The size of a text area is specified by the cols and rows attributes (or with CSS).
- The name attribute is needed to reference the form data after the form is submitted (if you omit the name attribute, no data from the text area will be submitted).
- The id attribute is needed to associate the text area with a label.

➤ Syntax:

<form>

Enter your address:

<textarea rows="2" cols="20"></textarea>

Enter your address:

<textarea></textarea>

</form>

- The rows attribute specifies the visible number of lines in a text area.
- The cols attribute specifies the visible width of a text area.

<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!

Tip: Always specify the type attribute for a <button> element, to tell browsers what type of button it is.

Example

A clickable button is marked up as follows:

```
<button type="button">Click Me!</button>
```

<fieldset>

- The <fieldset> tag is used to group related elements in a form.
- The <fieldset> tag draws a box around the related elements.

Syntax:

<form>

 <fieldset>

 <legend> write name which you want to give to your grouped elements </legend>

 <label> give name of element which is in group </label>

 <input type="write appropriate input type">

 </fieldset>

</form>

<legend>

- The <legend> tag defines a caption for the <fieldset> element.

Syntax:

<form>

<fieldset>

<legend> write name which you want to give to your grouped elements </legend>

<label> give name of element which is in group </label>

<input type="write appropriate input type">

</fieldset>

</form>

`<select>` (drop-down / pull-down list)

- The `<select>` element is used to create a drop-down list.
- 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 omit the name attribute, no data from the drop-down list will be submitted).
- The id attribute is needed to associate the drop-down list with a label.
- The `<option>` tags inside the `<select>` element define the available options in the drop-down list.

Syntax:

```
<label> </label>
<select name=" " id=" ">
  <option value=" " > </option>
  <option value=" " > </option>
</select>
```

```
<label> </label>
<select name=" " id=" ">
  <optgroup label=" ">
    <option value=" " > </option>
    <option value=" " > </option>
  </optgroup>
</select>
```

-
- The `<option>` tag defines an option in a select list.
 - `<option>` elements go inside a `<select>`, `<optgroup>`, or `<datalist>` element.

Note: The `<option>` tag can be used without any attributes, but you usually need the value attribute, which indicates what is sent to the server on form submission.

- The `<optgroup>` tag is used to group related options in a `<select>` element (drop-down list).
- If you have a long list of options, groups of related options are easier to handle for a user.

HTML Input Types

Here are the different input types you can use in HTML:

```
<input type="button">  
<input type="checkbox">  
<input type="color">  
<input type="date">  
<input type="datetime-local">  
<input type="email">  
<input type="file">  
<input type="hidden">  
<input type="image">  
<input type="month">  
<input type="number">
```

<input type="password">

<input type="radio">

<input type="range">

<input type="reset">

<input type="search">

<input type="submit">

<input type="tel">

<input type="text">

<input type="time">

<input type="url">

<input type="week">

Input Type Text:

`<input type="text">` defines a single-line text input field.

Input Type Password:

`<input type="password">` defines a password field.

Input Type Radio:

`<input type="radio">` defines a radio button.

Radio buttons let a user select ONLY ONE of a limited number of choices.

Input Type Checkbox:

`<input type="checkbox">` defines a checkbox.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.

Input Type Submit:

`<input type="submit">` defines a button for submitting form data to a form-handler.

The form-handler is typically a server page with a script for processing input data.

The form-handler is specified in the form's action attribute.

Input Type Reset:

`<input type="reset">` defines a reset button that will reset all form values to their default values

Input Type Button:

`<input type="button">` defines a button.

Input Type Color:

The `<input type="color">` is used for input fields that should contain a color.

Depending on browser support, a color picker can show up in the input field.

Input Type Date:

The `<input type="date">` is used for input fields that should contain a date.

Depending on browser support, a date picker can show up in the input field.

Input Type Datetime-local:

The `<input type="datetime-local">` specifies a date and time input field, with no time zone.

Depending on browser support, a date picker can show up in the input field.

Input Type Email:

The `<input type="email">` is used for input fields that should contain an e-mail address.

Depending on browser support, the e-mail address can be automatically validated when submitted.

Some smartphones recognize the email type, and add ".com" to the keyboard to match email input.

Input Type Image:

The `<input type="image">` defines an image as a submit button. The path to the image is specified in the `src` attribute.

Input Type File:

The `<input type="file">` defines a file-select field and a "Browse" button for file uploads.

Input Type Month:

The `<input type="month">` allows the user to select a month and year. Depending on browser support, a date picker can show up in the input field.

Input Type Hidden

The `<input type="hidden">` defines a hidden input field (not visible to a user).

A hidden field lets web developers include data that cannot be seen or modified by users when a form is submitted.

A hidden field often stores what database record that needs to be updated when the form is submitted.

Note: While the value is not displayed to the user in the page's content, it is visible (and can be edited) using any browser's developer tools or "View Source" functionality. Do not use hidden inputs as a form of security!

Input Type Number:

The `<input type="number">` defines a numeric input field.

You can also set restrictions on what numbers are accepted.

Input Type Range:

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 Search:

The `<input type="search">` is used for search fields (a search field behaves like a regular text field).

Input Type Tel:

The `<input type="tel">` is used for input fields that should contain a telephone number.

Input Type Time:

The `<input type="time">` allows the user to select a time (no time zone). Depending on browser support, a time picker can show up in the input field.

Input Type Url:

The `<input type="url">` is used for input fields that should contain a URL address.

Depending on browser support, the url field can be automatically validated when submitted.

Some smartphones recognize the url type, and adds ".com" to the keyboard to match url input.

Input Type Week:

The `<input type="week">` allows the user to select a week and year.

Depending on browser support, a date picker can show up in the input field.

HTML Input Attributes

The value Attribute:

The input value attribute specifies an initial value for an input field.

The size Attribute:

The input size attribute specifies the visible width, in characters, of an input field.

The default value for size is 20.

Note: The size attribute works with the following input types: text, search, tel, url, email, and password.

The height and width Attributes:

The input height and width attributes specify the height and width of an `<input type="image">` element.

The maxlength Attribute:

The input maxlength attribute specifies the maximum number of characters allowed in an input field.

Note: When a maxlength is set, the input field will not accept more than the specified number of characters. However, this attribute does not provide any feedback. So, if you want to alert the user, you must write JavaScript code.

The min and max Attributes:

The input min and max attributes specify the minimum and maximum values for an input field.

The min and max attributes work with the following input types: number, range, date, datetime-local, month, time and week.

Tip: Use the max and min attributes together to create a range of legal values.