

HTML Forms

.HTML <textarea> tag in form

.The <textarea> tag in HTML is used to insert multiple-line text in a form. The size of <textarea> can be specify either using "rows" or "cols" attribute or by CSS.

.<form>

- Enter your address:

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

list of HTML 5 form tags.

`<form>`- It defines an HTML form to enter inputs by the used side.

`<input>` -It defines an input control.

`<textarea>` -It defines a multi-line input control.

`<label>`-It defines a label for an input element.

`<fieldset>`-It groups the related element in a form.

`<legend>`-It defines a caption for a `<fieldset>` element.

`<select>`-It defines a drop-down list.

`<optgroup>`-It defines a group of related options in a drop-down list.

`<option>`-It defines an option in a drop-down list.

`<button>`-It defines a clickable button.

Label Tag in Form

- It makes the code parser/browser/user friendly.
- If you click on the label tag, it will focus on the text control. To do so, you need to have for attribute in label tag that must be same as id attribute of input tag.

`<form>`

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

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

`<label for="lastname">Last Name: </label>`

`<input type="text" id="lastname" name="lastname"/>`

`
`

`</form>`

HTML Password Field Control

- The password is not visible to the user in password field control.

```
<form>
```

```
  <label for="password">Password: </label>
```

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

```
</form>
```

HTML 5 Email Field Control

- The email field is new in HTML 5. It validates the text for correct email address. You must use @ and . in this field.

```
<form>
```

```
  <label for="email">Email: </label>
```

```
    <input type="email" id="email" name="email"/>
```

```
  <br/>
```

```
</form>
```

Radio Button Control

- The radio button is used to select one option from multiple options. It is used for selection of gender, quiz questions etc.
- If you use one name for all the radio buttons, only one radio button can be selected at a time.
- Using radio buttons for multiple options, you can only choose a single option at a time.

`<form>`

`<label for="gender">Gender: </label>`

`<input type="radio" id="gender" name="gender" value="male"/>Male`

`<input type="radio" id="gender" name="gender" value="female"/>Female
`

Checkbox Control

- The checkbox control is used to check multiple options from given checkboxes.

`<form>`

Hobby:`
`

```
    <input type="checkbox" id="cricket" name="cricket"
value="cricket"/>
```

```
    <label for="cricket">Cricket</label> <br>
```

```
    <input type="checkbox" id="football"
name="football" value="football"/>
```

```
    <label for="football">Football</label> <br>
```

```
    <input type="checkbox" id="hockey" name="hockey"
value="hockey"/>
```

Submit button control

- HTML `<input type="submit">` are used to add a submit button on web page. When user clicks on submit button, then form get submit to the server.
- Syntax:
`<input type="submit" value="submit">`
- The `type = submit` , specifying that it is a submit button
- The value attribute can be anything which we write on button on web page.

HTML <fieldset> element:

- The <fieldset> element in HTML is used to group the related information of a form.
- This element is used with <legend> element which provide caption for the grouped elements.

<form>

 <fieldset>

 <legend>User Information:</legend>

 <label for="name">Enter name</label>

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

 <label for="pass">Enter Password</label>

 <input type="Password" id="pass" name="pass">

HTML Form Input Types

- In HTML `<input type=" " >` is an important element of HTML form. The "type" attribute of input element can be various types, which defines information field. Such as `<input type="text" name="name">` gives a text box.

HTML Form Input Types

- **text** -Defines a one-line text input field
- **password** -Defines a one-line password input field
- **submit** -Defines a submit button to submit the form to server
- **reset** -Defines a reset button to reset all values in the form.
- **radio** -Defines a radio button which allows select one option.
- **checkbox**-Defines checkboxes which allow select multiple options form.
- **button**-Defines a simple push button, which can be programmed to perform a task on an event.

HTML5 added new types on `<input>` element

- **color** Defines an input field with a specific color.
- **date** Defines an input field for selection of date.
- **datetime-local** Defines an input field for entering a date without time zone.
- **email** Defines an input field for entering an email address.
- **Month** Defines a control with month and year, without time zone.
- **number** Defines an input field to enter a number.
- **url** Defines a field for entering URL
- **week** Defines a field to enter the date with week-year, without time zone.
- **search** Defines a single line text field for entering a string

<input type="reset">:

- The <input> type "reset" is also defined as a button but when the user performs a click event, it by default reset the all inputted values.

<form>

<label>User id: </label>

<input type="text" name="user-id" value="user">

<label>Password: </label>

<input type="password" name="pass" value="pass">

<input type="submit" value="login">

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

</form>

`<input type="image">`:

- The `<input>` type "image" is used to represent a submit button in the form of image.

HTML5 newly added <input> types element<input type="color">:

- The <input> type "color" is used to define an input field which contains a colour. It allows a user to specify the colour by the visual colour interface on a browser.

<form>

Pick your Favorite color:

<input type="color" name="upclick" value="#a52a2a">

Upclick

<input type="color" name="downclick" value="#f5f5dc">

Downclick

</form>

The default value of "color" type is #000000 (black). It only supports color value in hexadecimal format.

<input type="date">:

- The <input> element of type "date" generates an input field, which allows a user to input the date in a given format. A user can enter the date by text field or by date picker interface.

<form>

Select Start and End Date:

<input type="date" name="Startdate"> Start
date:

<input type="date" name="Enddate"> End date:

<input type="submit">

</form>

<input type="datetime-local">:

- The <input> element of type "datetime-local" creates input field which allow a user to select the date as well as local time in the hour and minute without time zone information.

<form>

<label>

Select the meeting schedule:

Select date & time: <input type="datetime-local"
name="meetingdate">

</label>

<input type="submit">

</form>

<input type="month">:

- The <input> type "month" creates an input field which allows a user to easily enter month and year in the format of "MM, YYYY" where MM defines month value, and YYYY defines the year value.
- <form>
- <label>Enter your Birth Month-year: </label>
- <input type="month" name="newMonth">
- <input type="submit">
- </form>

<input type="number">:

- The <input> element type number creates input field which allows a user to enter the numeric value. You can also restrict to enter a minimum and maximum value using min and max attribute.
- <form>
- <label>Enter your age: </label>
- <input type="number" name="num" min="50" max="80">
- <input type="submit">
- </form>

<input type="url">:

- The <input> element of type "url" creates an input field which enables user to enter the URL.
- <form>
- <label>Enter your website URL: </label>
- <input type="url" name="website"
 placeholder="http://example.com">

- <input type="submit" value="send data">
- </form>

`<input type="search">`:

- The `<input>` type "search" creates an input field which allows a user to enter a search string. These are functionally symmetrical to the text input type, but may be styled differently.
- `<form>`
- `<label>Search here:</label>`
- `<input type="search" name="q">`
- `<input type="submit" value="search">`
- `</form>`

HTML form Attribute

- HTML <form> element attributes

HTML action attribute

- The action attribute of <form> element defines the process to be performed on form when form is submitted, or it is a URI to process the form information.
- The action attribute value defines the web page where information proceed. It can be .php, .jsp, .asp, etc. or any URL where you want to process your form.
- If action attribute value is blank then form will be processed to the same page.

HTML method attribute

- The method attribute defines the HTTP method which browser used to submit the form. The possible values of method attribute can be:
- post: We can use the post value of method attribute when we want to process the sensitive data as it does not display the submitted data in URL.
- get: The get value of method attribute is default value while submitting the form. But this is not secure as it displays data in URL after submitting the form.

HTML target attribute

- The target attribute defines where to open the response after submitting the form. The following are the keywords used with the target attribute.
- `_self`: If we use `_self` as an attribute value, then the response will display in current page only.
- `_blank`: If we use `_blank` as an attribute it will load the response in a new page.

HTML autocomplete attribute

- The HTML autocomplete attribute is a newly added attribute of HTML5 which enables an input field to complete automatically. It can have two values "on" and "off" which enables autocomplete either ON or OFF. The default value of autocomplete attribute is "on".
- `<form action="action.html" method="get" autocomplete="on">`
- it can be used with `<form>` element and `<input>` element both.

HTML enctype attribute

- The HTML enctype attribute defines the encoding type of form-content while submitting the form to the server. The possible values of enctype can be:
- application/x-www-form-urlencoded: It is default encoding type if the enctype attribute is not included in the form. All characters are encoded before submitting the form.

```
<form action="action.html" method="post"  
  enctype="application/x-www-form-urlencoded" >
```

- multipart/form-data: It does not encode any character. It is used when our form contains file-upload controls.
- text/plain (HTML5): In this encoding type only space are encoded into + symbol and no any other special character encoded.

HTML novalidate attribute HTML5

- The novalidate attribute is newly added Boolean attribute of HTML5. If we apply this attribute in form then it does not perform any type of validation and submit the form.
- `<form action = "action.html" method = "get" novalidate>`

HTML <input> element attribute

- HTML name attribute

- The HTML name attribute defines the name of an input element. The name and value attribute are included in HTTP request when we submit the form.

- HTML value attribute

- The HTML value attribute defines the initial value or default value of an input field.

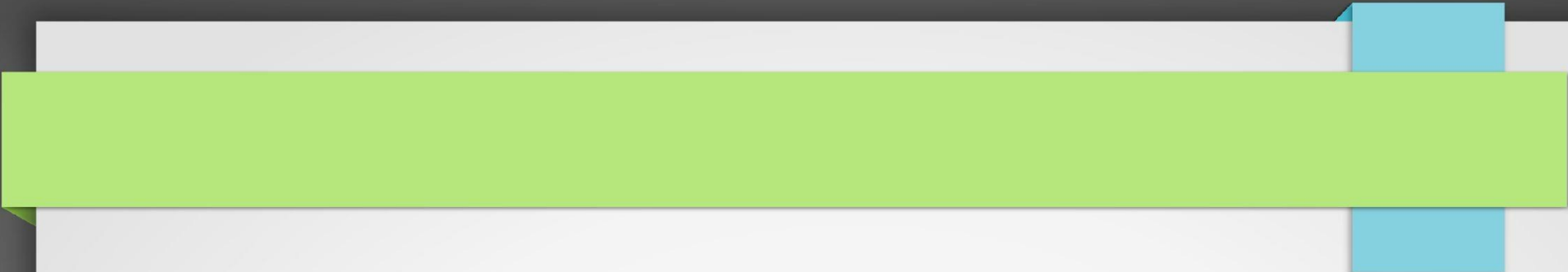
- HTML required attribute HTML5

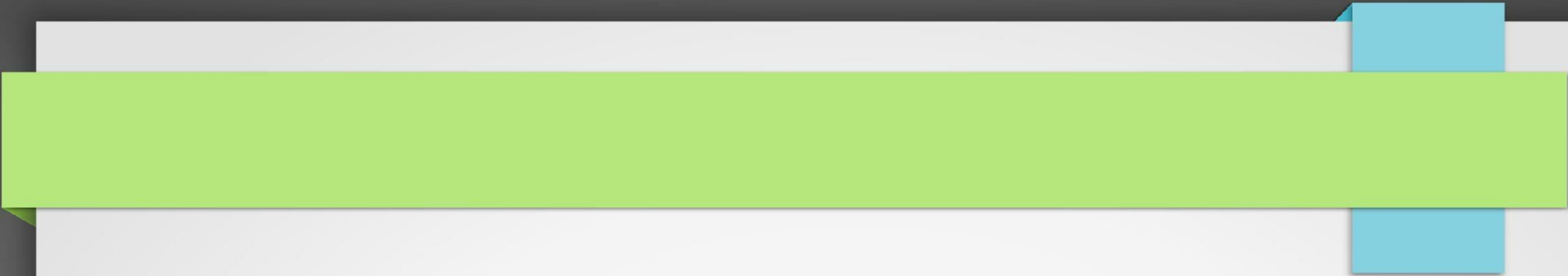
HTML required is a Boolean attribute which specifies that user must fill that field before submitting the form.

- **HTML autofocus attribute HTML5**

The autofocus is a Boolean attribute which enables a field automatically focused when a webpage loads. `<input type="text" name="uname" autofocus>

`

- 
- HTML disabled attribute-The HTML disabled attribute when applied then it disable that input field. The disabled field does not allow the user to interact with that field.
 - The disabled input filed does not receive click events, and these input value will not be sent to the server when submitting the form.
 - HTML size attribute-The size attribute controls the size of the input field in typed characters.

- 
- HTML placeholder attribute HTML5
 - The placeholder attribute specifies a text within an input field which informs the user about the expected input of that field.
 - The placeholder attribute can be used with text, password, email, and URL values.
 - When the user enters the value, the placeholder will be automatically removed.

HTML Audio

- The HTML `<audio>` element is used to play an audio file on a web page.

`<audio controls>`

`<source src="horse.ogg" type="audio/ogg">`

`<source src="horse.mp3" type="audio/mpeg">`

Your browser does not support the audio element.

`</audio>`

- The `controls` attribute adds audio controls, like play, pause, and volume.
- The `<source>` element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.

HTML Drag and Drop

- HTML Drag and Drop (DnD) is a feature of HTML5.
- It is a powerful user interface concept which is used to copy, reorder and delete items with the help of mouse.
- You can hold the mouse button down over an element and drag it to another location.
- If you want to drop the element there, just release the mouse button.
- If you want to achieve the Drag and Drop functionality in traditional HTML4, you must either have to use complex JavaScript programming or other JavaScript frameworks like jQuery etc.

Events for Drag and Drop feature

- **Event Description**
- **Drag** It fires every time when the mouse is moved while the object is being dragged.
- **Dragstart** It is a very initial stage. It fires when the user starts dragging object.
- **Dragenter** It fires when the user moves his/her mouse cursor over the target element.
- **Dragover** This event is fired when the mouse moves over an element.
- **Dragleave** This event is fired when the mouse leaves an element.
- **Drop** Drop It fires at the end of the drag operation.

HTML <audio> Autoplay

- To start an audio file automatically, use the autoplay attribute:

<audio controls autoplay>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

Your browser does not support the audio element.

</audio>

Add muted after autoplay to let your audio file start playing automatically (but muted):

Events for Drag and Drop feature

- **Event Description**
- **Drag** It fires every time when the mouse is moved while the object is being dragged.
- **Dragstart** It is a very initial stage. It fires when the user starts dragging object.
- **Dragenter** It fires when the user moves his/her mouse cursor over the target element.
- **Dragover** This event is fired when the mouse moves over an element.
- **Dragleave** This event is fired when the mouse leaves an element.
- **Drop** Drop It fires at the end of the drag operation.