Web Designing

Working with Forms, Images and Media in HTML5

Outline

- Working with Forms, Images and Media in HTML5
  - Inserting Image in a Web Page
  - Image Map
  - Introduction to Canvas
  - Exploring FORM Element
  - Observing Various INPUT Elements, Button Element, Progress Element, Meter Element, Output Element, Datalist Element, Textarea Element, Label Element, Select Element, Option Element, Optgroup Element, Submitting a Form
  - Details and Summary Element
  - Menu Element
  - Command Element
  - Time Element

Outline

- Working with Forms, Images and Media in HTML5
  - KBD Element
  - Video Element
  - Audio Element
  - Embed Element
  - Object Element
  - Figure and Figcaption Element.

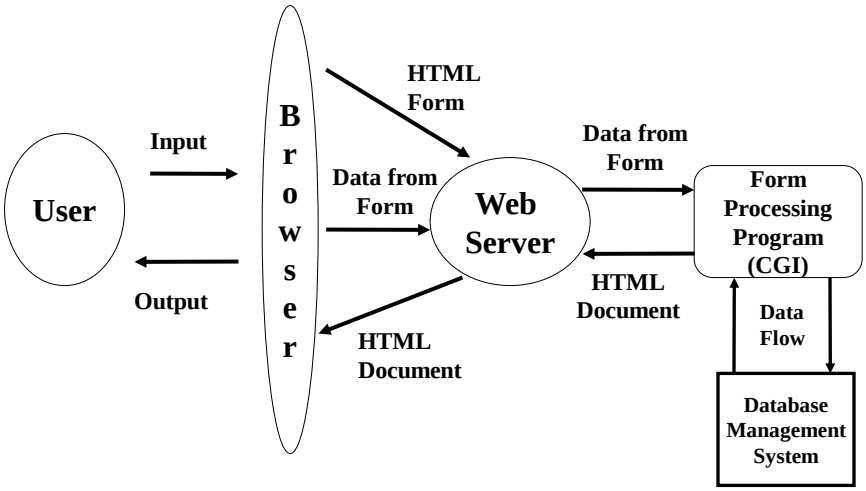
3

HTML Form

- Without forms, a Web site is "read-only" – it just provides information to the user
- Forms enable the user to provide information to the Web site. For example, the user can:
  - Perform searches on Web site
  - Give comments
  - Ask for info that is not available on the Website
  - Place order for goods and services

4

Form Processing



Flow of Information for Forms

HTML Form

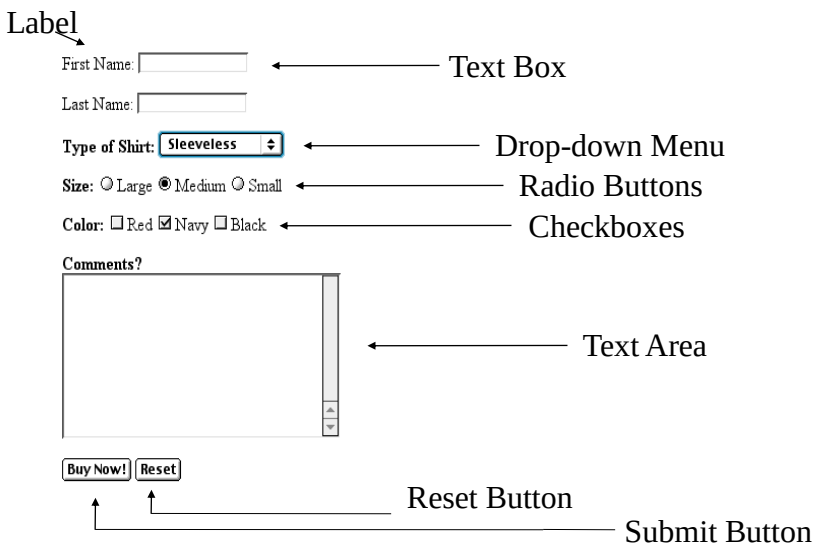
- Sometimes, you need to develop a website that require user interaction.
  - For example, you need to develop a railway website that allows users to check the arrival or departure status of trains on the Internet.
  - This website uses a registration or login form to collect the user information and submits the information to the server for further processing.
- These types of forms provides the medium of interaction between a website and its users on the Internet.

HTML Form

- A form is an area of a Web page that allows the users to provide their information in a variety of ways, such as by entering the text field or by selecting one or more available options from the provided list.
- HTML enables you to add a form in a Web page by using the FORM tag.
- Forms are always placed between the <BODY> and </BODY> tags of a Web page.
- After adding the form on the Web page, you can add various controls, such as buttons and text fields, on the form by using a variety of tags. Some examples of these elements are INPUT, BUTTON, TEXTAREA, and DATALIST.

7

Example: Form



Form Elements

- All form elements should be written in between the <form>..</form> tags.

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control
<label>	Defines a label for an <input> element
<textarea>	Defines a multiline input control
<select>	Defines a drop-down list
<option>	Defines an option in a drop-down list
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<button>	Defines a clickable button

Form Elements

- The INPUT tag allows you to enter different types of values, such as date, time, and e-mail address.
- The BUTTON tag enables you to add buttons, such as submit and cancel buttons on the form to submit or cancel form's detail.
- The TEXTAREA tag enables you to enter text in the provided area.
- The DATALIST tag allows you to enter the text in a text field by providing a set of predefined values.

10

Form Elements Attributes

- Most commonly used **FORM** element **Attributes**

Attribute	Description
method	Specifies the HTTP method used when submitting the form
action	Specifies an address (url) where to submit the form
autocomplete	Specifies if the browser should autocomplete the form
novalidate	Specifies that the browser should not validate the form.
name	Specifies a name used to identify the form
accept-character	Refers to the character set in the form that can be accepted by the server

11

Form Elements Attributes

- Most commonly used **FORM** element **Attributes**

Attribute	Description
enctype	Specifies how the information in the form should be encoded before sending it to the server
target	Opens the action URL in the specified target, such as in the same window, in a new window, or in a new tab

12

Form Elements Attributes

- The **METHOD** attribute specifies the HTTP method to be used when submitting the form data:
  - **GET**
  - **POST**

13

Form Elements Attributes

- **GET**;
  - The default method when submitting form data
  - Submitted form data will be visible in the page address field
  - The length of a URL is limited (about 3000 characters)
  - Never used to send sensitive data! Better for non-secure data
  - Useful for form submissions where a user want to bookmark the result

14

Form Elements Attributes

- **POST**;
  - The POST method does not display the submitted form data in the page address field.
  - Used for sensitive or personal information.
  - Has no size limitations, and can be used to send large amounts of data.

15

Form Elements Attributes

- The **ACTION** attribute defines the action to be performed when the form is submitted.
  - Normally, the form data is sent to a web page on the server when the user clicks on the submit button.
  - In the example below, the form data is sent to a page on the server called "action_page.php". This page contains a server-side script that handles the form data:
  - <form action="action_page.php">

16

Form Elements Attributes

- **AUTOCOMPLETE** attribute is used to provide an autocompletion option to user, when user visit the form page. Default value is "on“
  - If autocompletion is on, it will autocomplete the form and if autocompletion is off, the user have to fill the form field manual.
  - It is possible to have autocomplete “on” and “off” for the form, and “off” and “on” for specific input fields.
  - The autocomplete attribute works with <form> and the following <input> types:
    - text, search, url, tel, password, datepickers, color

```
<form action="demo_form.php" autocomplete="on">
```

17

Form Elements Attributes

- **NOVALIDATE** attribute is used to send the information for not validating the form field. It specifies that form data shouldn’t be validated.

  <form action="demo_form.php" novalidate>

- **NAME** attribute used to identify the form.
  - for DOM usage: (document.forms.name)

18

Form <input>

- The most important form element is the input element.
- You can create these controls by using the INPUT tag.
- An input element can vary in many ways, depending on the type attribute.
- A form is composed of controls, such as text box, drop-down list, check-box, and radio button that enable the user to enter the information.
- To do this, you need to set the type attribute of the INPUT tag to the name of the control.
- For example, <INPUT type="text"> creates a control of text box type.

19

Types of input tag

- text and search
- tel
- url
- email
- password
- datetime-local
- datetime, date, month, week, and time
- number and range
- file
- hidden
- checkbox
- radio
- submit
- reset

20

Types of input tag

- <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">

21

Types of input tag

- <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">
- Note: The default value of the type attribute is "text".

22

Input Element Attributes

- **TYPE** (required)
  - Defines the usage of the **INPUT** element.
  - Hidden inputs always have **TYPE = "hidden".**
- **NAME** provides a unique identification for **INPUT** element.
  - Each input field must have a **name** attribute to be submitted.
  - If the name attribute is omitted, the data of that input field will not be sent at all.
- **VALUE** indicates the value that the **INPUT** element sends to the server upon submission.
  - <input type="text" id="fname" name="fname" value="John">

23

Input Element Attributes

- **SIZE** attribute specifies the size the visible width, in characters, of an input field. The default value for size is 20 <input type="text" id="fname" name="fname" size="50">
  - <input type="text" id="fname" name="fname" size="50">
- **MAXLENGTH** attribute specifies the maximum number of characters that the input field will accept.
  - <input type="text" id="pin" name="pin" maxlength="4" size="4">

24

Input Element Attributes

- The **READONLY** attribute specifies that the input field is read only (cannot be changed)
  - <input type="text" id="fname" name="fname" value="John" readonly>
- The **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.
  - <label for="quantity">Quantity (between 1 and 5):</label>
    <input type="number" id="quantity" name="quantity" min="1" max="5">

25

Input Element Attributes

- The **DISABLED** attribute specifies that the input field is disabled.
  - A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form
  - <input type="text" name="fname" value="MCA" disabled>
- The **multiple** attribute specifies that the user is allowed to enter more than one value in an input field.
  - The multiple attribute works with the following input types: email, and file.
  - <label for="files">Select files:</label>
    <input type="file" id="files" name="files" multiple>

26

Input Element Attributes

- The **autofocus** attribute specifies that an input field should automatically get focus when the page loads.
  - <input type="text" id="fname" name="fname" autofocus>

27

Input Type

- **Text** allow the users to enter a single-line text such as name, date of birth, telephone number, or e-mail, from the user. Default width of a text field is 20 characters.

Example

```
First name: <input type="text" name="fname" size="25"><br>
Last name:  <input type="text" name="lname" size="25">
```

Result

First name:

Last name:

28

Working with input element

```
<!DOCTYPE HTML >
<HTML>
  <HEAD>
     <TITLE>input</TITLE>
   </HEAD>
   <BODY>
       <FORM>
          Enter your name:<INPUT>
       </FORM>
    </BODY>
</HTML>
```

29

Working text field

```
<!DOCTYPE HTML >
<HTML>
  <HEAD>
     <TITLE>textname</TITLE>
   </HEAD>
   <BODY>
       <FORM>
         Enter your username:<INPUT type="text" name=
          "username" >
       </FORM>
    </BODY>
</HTML>
```

30

Working text field for maxlength attribute

```
<!DOCTYPE HTML >
<HTML>

<HEAD>
<TITLE>textmaxlen</TITLE>
</HEAD>
<BODY>
<FORM>
Enter your 2 digit code number:<INPUT type=”text” name=
“username” size="2"     maxlength="2">
</FORM>

</BODY>
</HTML>
```

31

Working text field for default value

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>textvalue</TITLE>
</HEAD>
<BODY>
<FORM>
Enter your two digit code number:<INPUT type=”text” name=
“username” size="2"     maxlength="2" value="00">
</FORM>
</BODY>
</HTML>
```

32

Input Type password

- **Password** are like text boxes, except the characters in a password field are automatically masked. (shown as asterisks or circles)

Example

```
User Name:<br>
<input type="text" name="username"><br>
Password:<br>
<input type="password" name="pswd">
```

Result

User Name:

Password:

33

Input Type password

- A password is used to protect the secret information that a user does not want to share with others.
- Therefore, it is also referred as a masked text box.

34

Working with password

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>password</TITLE>
</HEAD>
<BODY>
<FORM>
Enter Password : <INPUT type="password">
</FORM>
</BODY>
</HTML>
```

35

Working with password field with name attribute

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>pwname</TITLE>
</HEAD>
<BODY>
<FORM>
Enter Password : <INPUT type="password“ name=“pw”>
</FORM>
</BODY>
</HTML>
```

36

Working with password field with size attribute

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>pwsize</TITLE>
</HEAD>
<BODY>
<FORM>
Enter Password : <INPUT type="password“ name=“pw” size =“10”>
</FORM>
</BODY>
</HTML>
```

37

Working with password field with maxlength attribute

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>pwmaxlen</TITLE>
</HEAD>
<BODY>
<FORM>
Enter Password : <INPUT type="password“ name=“pw” size =“10”
maxlenth=“4”>
</FORM>
</BODY>
</HTML>
```

38

Working with password field with value attribute

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>pwvalue</TITLE>
</HEAD>
<BODY>
<FORM>
Enter Password : <INPUT type="password“ name=“pw” size =“10”
maxlenth=“4” value=“dreamtech”>
</FORM>
</BODY>
</HTML>
```

39

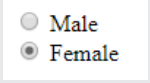
Input Type radio

- **Radio** : Usually found in a group of options, only one radio button in a group can be selected at a time.
- Selecting one radio button deselects the others in its group.
- Each radio button within a group should have the same name and different values. (Otherwise, browsers cannot distinguish between them)
- **CHECKED** attribute indicates which radio button is selected initially

Example

```
<input type="radio" name="gender" value="male"> Male<br>
<input type="radio" name="gender" value="female" checked>Female<br>
```

Result



Creating radio button field

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>radio button</TITLE>
</HEAD>
<BODY>
<FORM>
Select one of the three choices:<BR/>
<INPUT type="radio" name="group1" value="Milk"> Milk<BR>
<INPUT type="radio" name="group1" value="Butter" checked>
Butter<BR>
<INPUT type="radio" name="group1" value="Cheese"> Cheese
</FORM>
</BODY>
</HTML>
```

41

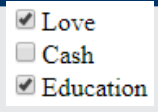
Input Type checkbox

- **Check** boxes let a user select NONE/ONE/MORE options of a limited number of choices.
- Each check box within a group should have the same name and different values. (Otherwise, browsers cannot distinguish between them)
- **CHECKED** attribute indicates initially selected checkboxe/s.

Example

```
<input type="checkbox" name="choice" value="cb1" checked>Love <br>
<input type="checkbox" name="choice" value="cb2">Cash <br>
<input type="checkbox" name="choice" value="cb3" checked>Education <br>
```

Result



Creating checkbox field

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>checkbox</TITLE>
</HEAD>
<BODY>
<FORM>
Select one of the three choices:<BR/>
<INPUT type="checkbox" name="option1" value="Milk">
Milk<BR>
<INPUT type="checkbox" name="option2" value="Butter"
checked="checked"> Butter<BR>
<INPUT type="checkbox" name="option3" value="Cheese">
Cheese<BR>
</FORM>
</BODY>
</HTML>
```

43

Input Type hidden

- A **hidden** field is used to pass along variables and values from one form to another, without forcing the user to re-enter the information.
- In addition, it is not displayed by the browser. You can create a hidden field by using the INPUT tag and setting its type attribute to **hidden.**
- **Value** - Specifies the value that is to be displayed by default in the hidden field

44

Creating hidden field

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>Using Hidden fields</TITLE>
</HEAD>
<BODY>
<FORM>
User name:<INPUT id="name" type="text"/><BR/>
<INPUT type="hidden" id="country" value="India" /><BR/>
<INPUT type="submit" onclick="alert('Hello, ' +
document.getElementById('name').value + ' from '+
document.getElementById('country').value)" value="Submit">
</FORM>
</BODY>
</HTML>
```

45

Input Type submit

- **Submit button** : <input type="submit"> defines a submit button.
- A submit button is used to send form data to a server.
- The data is sent to the page specified in the form's action attribute.
- The file (form-handler) defined in the action attribute usually does something with the received input. (include script for processing input data).
- **VALUE** attribute changes the text displayed on the button (default is "**Submit**").

46

Input Type submit

Example

```
<form name="input" action="html_form_action.asp" method="get">
Username: <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

Result

Username: Submit

If you type some characters in the text field above, and click the "Submit" button, the browser will send your input to a page called "html_form_action.asp".

47

Creating submit button field

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>submit button</TITLE>
</HEAD>
<BODY>
<FORM><BR>
username:<INPUT type="text" name="username"
value="dreamtech"> <BR/>
password:<INPUT type="password" name="password"
value="kogent"> <BR/>
<INPUT type="submit" name=”submit” value="click!"><BR/>
</FORM>
</BODY>
</HTML>
```

48

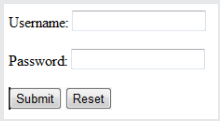
Input Type reset

- A **reset button** is used to clear all the entries user entered into the form and reset the form-data to its default values.
- **VALUE** attribute changes the text displayed on the button (default is "**Reset**")

Example

```
<form name="input" action="html_form_action.asp" method="get">
<P>Username: <input type="text" name="user" size="25"></P>
<P>Password: <input type="password" name="pswd" size="25"></P>
<P><input type="submit" value="Submit">
<input type="reset" value="Reset"></P></form>
```

Result



49

Creating reset button field

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>reset button</TITLE>
</HEAD>
<BODY>
<FORM><BR>
username:<INPUT type="text" name="username"
value="dreamtech"><BR/>
password:<INPUT type="password" name="password"
value="kogent"><BR/>
<INPUT type="submit" name=”submit” value="click!">
<INPUT type="reset" name=”reset” value="reset">
</FORM>
</BODY>
</HTML>
```

50

Form Element <label>

- The **<label>** tag defines a label for an **<input>** element.
- Some controls, such as button control, do not require any description as they already have labels associated with them, whereas, some controls, such as text boxes, check boxes, and radio buttons, need description.
- The <label> element does not render as anything special for the user. However, it provides a usability improvement for mouse users, because if the user clicks on the text within the <label> element, it toggles the control.

51

Form Element <label>

- Each LABEL is associated with exactly one control.
- The **for** attribute of the <label> tag should be equal to the **id** attribute of the related element to bind them together.
- A label can be bound to an element either by using the "for" attribute, or by placing the element inside the <label> element.

52

Form Element <label>

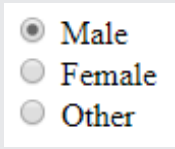
Example

```
<input type="radio" name="gender" id="male" value="male" checked>
<label for="male">Male</label><br>

<input type="radio" name="gender" id="female" value="female">
<label for="female">Female</label><br>

<input type="radio" name="gender" id="other" value="other">
<label for="other">Other</label><br>
```

Result



53

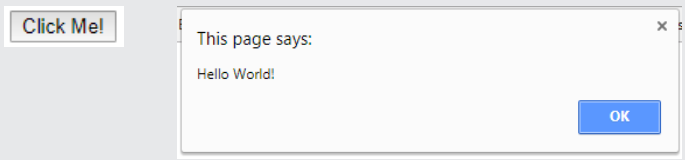
Form Element <button>

- The <button> element defines a clickable button.

Example

```
<button type="button" onclick="alert('Hello World!')">
Click Me!
</button>
```

Result



54

Form Element <button>

- A button control can be used to perform various tasks, such as submitting or resetting the details of the form.
- The BUTTON tag can be used with the conjunction of FORM tag to display the controls on the form.
- A button control can also be placed on the form by using the INPUT tag, but the difference is that you do not have the scope to change the appearance of the button control, except changing the text on the button control.
- However, you can change the appearance of the button control in case you have created the button control by using the BUTTON tag.

55

Form Element <button>

- The button control is created by using the opening and the closing tags of the BUTTON tag.
- The text, image, or any multimedia embedded between the opening and the closing tags of the BUTTON tag become the content of the button control.
- The BUTTON tag provides a type attribute that allows you to create three kinds of button controls i.e. submit button, reset button, and normal button.
- The submit button is used to submit the form, whereas the reset button is used to erase all the text entered in the text box of the form and set the default values.
- The submit button is created by setting the type attribute to submit, the reset button is created by setting the type attribute to reset, and the normal button

56

<button> attributes

Button Tag Attributes	Description
autofocus	Allows the button control to get the focus as soon as page loads
disabled	Disables the button control
form	Refers to the id of the FORM tag
formaction	Refers to the value of the action attribute of the current form
formenctype	Specifies a value that is used to encode the content while submitting to the server. The possible values are application/x-www-form-urlencoded (default), multipart/form-date, text/plain
formmethod	Specifies the methods of the HTTP at the time of the submitting the button control. The possible values are get, post, put, and delete

57

<button> attributes

Button Tag Attributes	Description
formnovalidate	Specifies that the form is not to validate at the time of submitting the button control
formtarget	Specifies the destination, such as a new tab or a new window, to load the browsing context
name	Provides a name of the button control
type	Specifies the type of the button control. The possible values are submit, reset, and button
value	Provides a value to the button control. You can use this attribute if the form attribute is present

58

Using button element

```
<!DOCTYPE HTML>
<HTML>
        <HEAD>
                <TITLE> Using the BUTTON element</TITLE>
        </HEAD>
        <BODY>
                <P>Simple Submit Button</P>
                <BUTTON type="submit">SUBMIT</BUTTON>
                <HR/>
                <P>Changing the font of the button text.</P>
                <BUTTON type="submit" style="color:red;font-
                        size:20px;" >
                      <B>SUBMIT</B>
                </BUTTON>

        </BODY>
</HTML>
```

59

Using button element to display image

```
<!DOCTYPE HTML>
<HTML>
        <HEAD>
                <TITLE>Using the BUTTON element</TITLE>
        </HEAD>
        <BODY>
                <P>Displaying Image on Button </P>
                <BUTTON type="submit" >
                        <IMG src="back.png" width="50"
                          height="50">
                </BUTTON>
                <BUTTON type="submit" >
                        <IMG src="next.png" width="50"
                           height="50">
                </BUTTON>
        </BODY>
</HTML>
```

60

Form Element

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

61

Input type

- <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">

62

Input type

- <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">

63

Form Element <textarea>

- Inserts a scrollable text box into FORM for entering multi-line text.
- It is commonly used in situations where you ask for info that may require multiple sentences.
- The content provided within the starting and the ending tags of the TEXTAREA tag should only be plain text.
- You control the dimension of the text area by using the **ROWS** and **COLS** attributes.
- The **rows** attribute specifies the visible number of lines in a text area.
- The **cols** attribute specifies the visible width of a text area.

64

Form Element <textarea>

- The wrap attribute of the TEXTAREA tag defines how the text appears in the textarea field when it reaches the end of every row. Wrapping can have **soft, hard,** and **off settings**.
  - The soft setting forces the words to wrap inside the textarea, but when the form is submitted, it includes the line breaks.
  - The hard setting wraps the words inside the textarea and places line breaks at the end of each line so that when the form is submitted, it appears exactly as it appears in the textarea.
  - The offsetting ignores all the wrapping in the textarea and places the text into one ongoing line.

65

Form Element <textarea>

Textarea Attribute	Description
cols	Refers to the visible width of the textarea control
rows	Refers to the permitted number of rows in the textarea control
disabled	Disables the textarea
name	Refers to the name of the textarea
readonly	Specifies that the textarea is read-only and you cannot write in it
accesskey	Refers to the shortcut key on the keyboard
autofocus	Allows the control to get the focus as soon as the page loads

66

Form Element <textarea>

Textarea Attribute	Description
dirname	Specifies the name of the input control that indicates the text direction of the textarea
maxlength	Specifies the maximum number of character that can be entered in the textarea
placeholder	Helps the user to fill the respective textarea by providing the hint for the input
required	Specifies that the value of the input field is required to submit the form
wrap	Allows the textarea to wrap the text

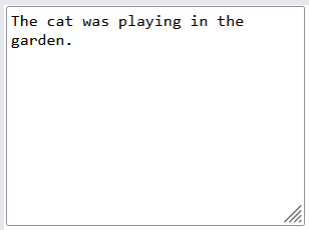
67

Form Element <textarea>

Example

```
<textarea name="message" rows="5" cols="30">
The cat was playing in the garden.</textarea>
```

Result



68

Working with textarea

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>textarea</TITLE>
</HEAD>
<BODY>
<FORM>
<TEXTAREA>
Enter your text here
</TEXTAREA>
</FORM>
</BODY>
</HTML>
```

69

Form Element <select>,<option>

- Multiple-Choice tags offer multiple choices to the user in a Web page, such as check boxes and radio buttons.
- **List Box**
- **<select>** tag presents a drop-down list with choices indicated by the **<option>** tags
- This is divided into the following three parts: Select Tag, Option Tag, Optgroup Tag
- Include **NAME** attribute
- 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
- Change the number of list options visible by including the **SIZE** = "x" attribute inside the <SELECT> tag
  - *x* number of options visible

70

Form Element <select>,<option>

- Use the **multiple** attribute to allow the user to select more than one value:
  - if you use multiple attribute, you should also assign different values for each of the **value** attributes of option tags
- The SELECT tag allows the user to select a single item from number of options.
- Unlike the radio button, the SELECT tag does not provide any default option that is to be transmitted to the server when no option is selected by the user.
- The OPTGROUP tag is used to create nested and cascading drop-down lists, the related items are grouped under specific headings. The attributes of the OPTGROUP tag are:
  - **Label** - Refers to the heading of the several groups in the cascading menu
  - **Disabled** - Disables the OPTGROUP tag

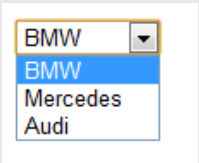
71

Form Element <select>,<option>

Example

```
<select name="cars">
<option selected>BMW</option>
<option>Mercedes</option>
<option>Audi</option>
</select>
```

Result



72

Using the select and option element

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>select</TITLE>
</HEAD>
<BODY>
<FORM>
Select any one fruit from the drop-down list<BR/>
<SELECT>
  <OPTION value="mango">mango</OPTION>
  <OPTION value="orange">orange</OPTION>
  <OPTION value="apple">apple</OPTION>
  <OPTION value="banana">banana</OPTION>
</SELECT>
</FORM>
</BODY>
</HTML>
```

73

Using the optgroup element

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>optgroup</TITLE>
</HEAD>
<BODY>
<FORM>
Select one item from a list of fruits, vegetables, and trees:
<SELECT>
<OPTGROUP label=”fruits”>
  <OPTION value="mango">mango</OPTION>
  <OPTION value="orange">orange</OPTION>
  <OPTION value="apple">apple</OPTION>
 </OPTGROUP>
```

74

Using the optgroup element

```
<OPTGROUP label=”Vegetables”>
  <OPTION value="cabbage">cabbage</OPTION>
  <OPTION value="tomato">tomato</OPTION>
  <OPTION value="potato">potato</OPTION>
 </OPTGROUP>
<OPTGROUP label=”trees”>
  <OPTION value="deodar">deodar</OPTION>
  <OPTION value="teak">teak</OPTION>
  <OPTION value="banyan">banyan</OPTION>
 </OPTGROUP>
</SELECT>
</FORM>
</BODY>
</HTML>
```

75

Using the select multiple and option element

```
<!DOCTYPE html>
<html>
<body>

<h1>The select multiple attribute</h1>

<p>The multiple attribute specifies that multiple options can be selected at once:</p>

<form action="/action_page.php">
  <label for="cars">Choose a car:</label>
  <select name="cars" id="cars" multiple>
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
    <option value="opel">Opel</option>
    <option value="audi">Audi</option>
  </select>
```

76

Using the select multiple and option element

```
<br><br>
  <input type="submit" value="Submit">
</form>

<p>Hold down the Ctrl (windows) or Command (Mac) button to select multiple options.</p>

</body>
</html>
```

- **Note :** Selecting multiple options vary in different operating systems and browsers:
- For windows: Hold down the control (ctrl) button to select multiple options
- For Mac: Hold down the command button to select multiple options
- You have to inform the user that multiple selection is available, it is more user-friendly to use checkboxes instead.

77

Form Element <fieldset>,<legend>

- **Grouping Form Data**
- The **<fieldset>** element is used to group related data in a single box.
- The **<legend>** element defines a caption for the **<fieldset>** element.

Example

```
<fieldset><legend>Personal Information:</legend>
    Name:<br>
    <input type="text" name="firstname" value="your first name"><br>
    Surname:<br>
    <input type="text" name="lastname" value="your last name">
</fieldset>
```

Result

Personal Information:
Name:
your first name
Surname:
your last name

78

Using fieldset and legend element

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>grouping</TITLE>
</HEAD>
<BODY>
<FORM>
<FIELDSET>
<LEGEND>  Login window </LEGEND>
<LABEL>    username<INPUT type=”text”></LABEL>  <BR/>
<LABEL> password<INPUT type=”password”>  </LABEL> <BR/>
<LABEL> <INPUT type="submit" name=”submit” value="click!"> </LABEL>
</FIELDSET>
</FORM>
</BODY>
</HTML>
```

79

Form Example

Feedback Form

Please fill out this form to help us improve our site.

Name:

Comments:

E-mail Address:

Things you liked:
Site design ☐ Links ☐ Ease of use ☐ Images ☐ Source code ☐

How did you get to our site?:
Search engine ◉ Links from another site ○ Reference in a book ○ Other ○

Rate our site: 10 ▾

Submit Clear

80

Form Example

```
<!DOCTYPE html>

<!-- Form using a variety of components. -->
<html>
  <head>
    <meta charset = "utf-8">
    <title>Form Example-1</title>
  </head>
  <body>
     <h1>Feedback Form</h1>
     <p>Please fill out this form to help us improve our site.</p>

     <form method = "post" action = "">

        <p><label><strong>Name:</strong>
           <input name = "name" type = "text" size = "25">
        </label></p>
```

81

Form Example

```
    <p><label><strong>Comments:</strong><br>
         <textarea name = "comments" rows = "4" cols =
          "36"></textarea>
      </label></p>

      <p><label><strong>E-mail Address:</strong>
          <input name = "email" type = "email" size = "25">
       </label></p>

       <p><strong>Things you liked:</strong><br>
           <label>Site design
              <input name = "thingsliked" type = "checkbox" value
            = "Design"></label>
           <label>Links
              <input name = "thingsliked" type = "checkbox" value
            = "Links"></label>
          <label>Ease of use
             <input name = "thingsliked" type = "checkbox" value =
          "Ease"></label>
```

82

Form Example

```
  <label>Images
     <input name = "thingsliked" type = "checkbox" value =
   "Images"></label>
   <label>Source code
       <input name = "thingsliked" type = "checkbox" value =
   "Code"></label>
</p>

<p><strong>How did you get to our site?:</strong><br>
    <label>Search engine
        <input name = "howtosite" type = "radio" value =
    "search engine" checked></label>
    <label>Links from another site
          <input name = "howtosite" type = "radio" value =
      "link"></label>
    <label>Reference in a book
          <input name = "howtosite" type = "radio" value =
      "book"></label>
   <label>Other
      <input name = "howtosite" type = "radio" value = "other"></label>
</p>
```

83

Form Example

```
  <p>
    <b>Rate our site:<b>
    <select name = "rating">
       <option selected>10</option>
       <option>9</option>
       <option>8</option>
       <option>7</option>
       <option>6</option>
       <option>5</option>
       <option>4</option>
       <option>3</option>
       <option>2</option>
       <option>1</option>
    </select>
 </p>
```

84

Form Example

```
        <p>
            <input type = "submit" value = "Submit">
            <input type = "reset" value = "Clear">
        </p>
      </form>
  </body>
</html>
```

85

Form Element <datalist>

- The **DATALIST** tag is used to display the list of the predefined options that the user may want to select as input.
- This tag enables the auto complete feature on the forms. This means that when a user start typing in a text box, a list of predefined words is dropped down to choose.
- The DATALIST tag is used with the **INPUT** tag, in which the **list** attribute is specified.
- The **value** of the **list** attribute is similar to the **id** attribute of the DATALIST tag to link the INPUT tag with the DATALIST tag.

86

Form Element <datalist>

- The OPTION tag used as the child tag of the DATALIST tag is used to specify the list of the options that are to be displayed.
- Example

```
<INPUT type="text" list="name">
<DATALIST id="name">
      <OPTION value="Richard">
      <OPTION value="John">
      <OPTION value="Tony">
</DATALIST>
```

87

Form Element <datalist>

- Here, we have used the INPUT tag and the DATALIST tag. We have specified the same value for the list attribute of the INPUT tag and the id attribute of the DATALIST tag.
- We have also defined the OPTION tags and specified their value attributes as the options to be displayed.

88

Using datalist element

```
<!DOCTYPE HTML>
<HTML>
    <HEAD>
        <TITLE>Using the DataList Element</TITLE>
    </HEAD>
    <BODY>
        <P>Enter the name of the your favorite car :</P>
        <INPUT type="text" name="favCar" list="cars">
        <DATALIST id="cars">
            <OPTION value="BMW">
            <OPTION value="Porsche">
            <OPTION value="Audi">
            <OPTION value="Ford">
```

89

Using datalist element

```
            <OPTION value="Ferrari">
            <OPTION value="Mercedes">
        </DATALIST>
    </BODY>
</HTML>
```

90

Form Element <output>

- The **OUTPUT** tag is used to display the result of the calculation, which can be written using the JavaScript.
- The OUTPUT tag has three attributes i.e. **form**, **name**, and **for**.
- The **form** attribute is used to specify the name of the form in which the output is displayed.
- The **name** attribute is used to specify the name of the current tag and the for attribute is used to specify the name of the control in which the result is displayed.
- Following is an example of using the OUTPUT tag.

```
<FORM>
    <OUTPUT name="result">
    </OUTPUT>
</FORM>
```

91

Form Element <output>

- **OUTPUT** Tag Attributes
  - **For**-- Associates the output with a specific control. The value of this attribute must match the id attribute of its associated control.
  - **Form**--Refers to the id of a form
  - **Name**-- Specifies the name of the OUTPUT tag

92

Using output element

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>Using the OUTPUT Element</TITLE>
        <SCRIPT type="text/javascript">
                function add()
                {
        document.forms["form"]["resultadd"].value=23+123;
                }
        </SCRIPT>
</HEAD>
```

93

Using output element

```
<BODY onload="add()">
        <H1>OUTPUT Element.</H1>
        <FORM name="form">
                Adding the two numbers:
                <OUTPUT name="resultadd">
                </OUTPUT>
        </FORM>
        </BODY>
</HTML>
```

94

Using output element to take input from user

```
<!DOCTYPE HTML >
<HTML>
<HEAD>
<TITLE>Using the OUTPUT Element</TITLE>
        <SCRIPT type="text/javascript">
                function add()
                {
                num1=parseInt(prompt("Enter first number.",0));
                num2=parseInt(prompt("Enter second number.",0));

        document.forms["form"]["resultadd"].value=num1+num2;
                }
        </SCRIPT>
</HEAD>
```

95

Using output element to take input from user

```
<BODY onload="add()">
        <H1>OUTPUT Element.</H1>

        <FORM name="form">
                Adding the two numbers:
                <OUTPUT name="resultadd">
                </OUTPUT>
        </FORM>
        </BODY>
</HTML>
```

96

<keygen>

- The **KEYGEN** tag is used to generate the key pair.
- When a form is submitted, a key pair, which contains the private and public keys, is generated using the KEYGEN tag to secure the content of the form.
- The private key from the generated key pair is encrypted and stored in the key database on local computer.
- The public key is encrypted and submitted to the server along with the form.
- Following is the example of the KEYGEN tag, as shows in the following code snippet.

<KEYGEN name="key_name“ challenge="987621“>

97

<keygen>

Keygen Attribute	Description
autofocus	Allows the control to get the focus as soon as the page loads
challenge	Specifies a string that is used for the verification at the time of submission of a form
disabled	Disables the input control
form	Refers to the id of the FORM tag
keytype	Specifies the type of the key to generate
name	Provides a name to the input control

98

Using keygen element

```
<!DOCTYPE HTML>
<HTML>
        <HEAD>
                <TITLE>Using the KEYGEN element</TITLE>
        </HEAD>
        <BODY>
                <FORM action="keygen.html" method="post"
enctype="text/plain">
                        First name: <INPUT type="text"
name="fname" /><BR />
                        Last name: <INPUT type="text"
name="lname" /><BR />
                        <KEYGEN name="key"
challenge="0987654321" keytype="RSA">
                        <INPUT type="submit" value="Submit"
/>
                </FORM>     </BODY>    </HTML>
```

99

<progress>

- The **PROGRESS** tag is used to display the progress of a particular task that is being performed.
- The PROGRES tag is used in conjunction with JavaScript to display the progress or the process of a task.
- The PROGRES tag has only two attributes i.e. **value** and **max**.
- The value attribute is used to specify how much the task has been completed and the max attributes is used to specify the total progress to be made.

100

<progress>

- PROGRESS Tag Example

```
<PROGRESS value="100" max="500">
</PROGRESS>
```

- Attributes used within the PROGRESS tag
  Value -- Specifies the value of the PROGRESS tag
  Max --Specifies the maximum value of the PROGRESS tag

101

Using progress element

```
<!DOCTYPE HTML>
<HTML>
    <HEAD>
        <TITLE>Using the PROGRESS Element</TITLE>
    </HEAD>
    <BODY>
        Current task is
        <PROGRESS  value="200" max="500">
        </PROGRESS>
        completed.
    </BODY>
</HTML>
```

102

Using progress element

```
<!DOCTYPE HTML>
<HTML>
        <HEAD>
                <TITLE>Using the KEYGEN element</TITLE>
        </HEAD>
        <BODY>
                <progress max=10 value=0>0%</progress>
                <progress max=100 value=25>25%</progress>
                <progress max=8 value=4>50%</progress>
                <progress max=100 value=75>75%</progress>
                <progress max=25 value=25>100%</progress>
               <progress>Unknown</progress>
</BODY>
</HTML>
```

103

<meter>

- The **METER** tag is used to define the scalar measurement.
- This tag is mostly useful when you need to display the disk usage and the relevance of a search result, or to show some other measurement.
- You cannot use the METER tag to display a single number as it is used to display a range.
- **METER Tag Example**

```
<METER min="50" max="165"> 72 </METER>
<METER>200 from 500</METER>
```

104

<meter>

- **METER Tag Example**
  <METER min="50" max="165"> 72 </METER>
  <METER>200 from 500</METER>
- Here, in the first example, we have used the min and max attributes of the METER tag to specify the minimum and maximum values for the tag.
- This implies that you cannot define the value of the METER tag less than 0 and more than 10.
- And in the second example, we have displayed the tag without any attribute.

105

<meter>

Meter tag Attribute	Description
Value	Specifies the value of the METER tag
min	Specifies the minimum value of the METER tag
max	Specifies the maximum value of the METER tag
low	Specifies a range which is considered as low value
high	Specifies a range which is considered as high value
optimum	Specifies a range which is considered as the optimum or the best value
form	Refers to the id of the FORM tag

106

Using meter element

```
<!DOCTYPE HTML>
<HTML>
        <HEAD>
                <TITLE>Using the METER element</TITLE>
        </HEAD>
        <BODY>
                <METER value="5" min="0" max="10"
style="width:200px"></METER> Passing
                Score <BR />
                <METER value="7" min="0" max="10"
style="width:200px"></METER>Your
                score<BR />
        </BODY>
</HTML>
```

107

HTML5 new input types

- color
- date
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week
- New input types that are not supported by older web browsers, will behave as <input type="text">.

108

Input types : color

- Is used for input fields that should contain a color.
- The color tool allows the user to choose a color using standard web formats.
- Depending on browser support, a color-picker can show up in the input field like the ones
- Syntax

```
<input type="color">
```

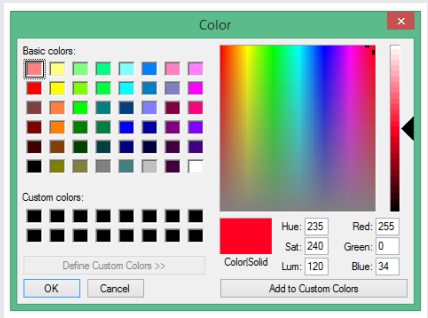
109

Input types : color

Example

```
Select your favorite color:
<input type="color" name="favcolor" value="#ff0022">
```

Result



110

Input types : color example

```
<!DOCTYPE html>
<html>
<body>
<h2>Show a Color Picker</h2>
 <p>The <strong>input type="color"</strong> is used for input fields that should contain a color.</p>
<form action="/action_page.php">
 <label for="favcolor">Select your favorite color:</label>
 <input type="color" id="favcolor" name="favcolor" value="#ff0000“>
```

111

Input types : color example

```
<input type="submit" value="Submit">
</form>
<p><b>Note:</b> type="color" is not supported in Internet Explorer 11 or Safari 9.1 (or earlier).</p>
</body>
</html>
```

112

Input types : date

- Setting the input type to date indicates that you wish the user to enter a date.
- Depending on browser support, a date picker can show up in the input field.
- You can restrict the dates allowed to a specific range by applying the min and max attributes to the element.
- Syntax

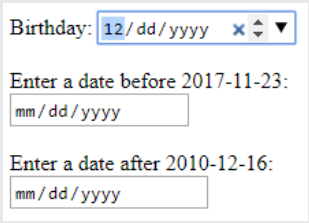
  <input type="date“>

Input types : date

Example

```
Birthday:
<input type="date" name="bday"><br><br>
Enter a date before 2017-11-23:<br>
<input type="date" name="bday" max="2017-11-22"><br><br>
Enter a date after 2010-12-16:<br>
<input type="date" name="bday" min="2010-12-17"><br><br>
```

Result



Input types : date example

```
<!DOCTYPE html>
<html>
<body>
<h2>Date Field</h2>
<p>The <strong>input type="date"</strong> is used for
   input fields that should contain a date.</p>
<form action="/action_page.php">
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
  <input type="submit" value="Submit">
</form>
```

115

Input types : date example

```
<p><strong>Note:</strong> type="date" is not supported
   in Internet Explorer 11 or prior Safari 14.1.</p>


</body>
</html>
```

116

Input types : date example min and max value

```
<!DOCTYPE html>
<html>
<body>
<h2>Date Field Restrictions</h2>
<p>Use the min and max attributes to add restrictions to
  dates:</p>
<form action="/action_page.php">
  <label for="datemin">Enter a date after 2000-01-
   01:</label>
  <input type="date" id="datemin" name="datemin"
   min="2000-01-02"><br><br>
```

117

Input types : date example min and max value

```
  <label for="datemax">Enter a date before 1980-01-
   01:</label>
  <input type="date" id="datemax" name="datemax"
   max="1979-12-31"><br><br>
  <input type="submit" value="Submit">
</form>
<p><strong>Note:</strong> type="date" is not supported
  in Internet Explorer 11 or prior Safari 14.1.</p>

</body>
</html>
```

118

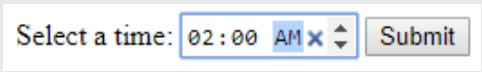
Input types : time

- The purpose of the time input type is to allow the user to enter a time.
- Depending on browser support a time picker might pop-up when you enter the input field.
- Syntax : <input type="time">

Example

```
Select a time:
<input type="time" name="times">
```

Result



119

Input types : time

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
<body>
<h2>Time Field</h2>
<p>The <strong>input type="time"</strong> is used for
   input fields that should contain a time.</p>
<form action="/action_page.php">
  <label for="tt">Select time:</label>
  <input type="time" id="tt" name="tt">
```

120

Input types : time

```
<input type="submit" value="Submit">
</form>
</body>
</html>
```

121

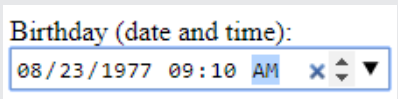
Input types : datetime-local

- The datetime-local element combines date and time in a single input field, with no time zone.
- Depending on browser support a time picker might pop-up when you enter the input field.
- Syntax : <input type="datetime-local">

Example

```
Birthday (date and time):<br>
<input type="datetime-local" name="bdaytime">
```

Result



122

Input types : datetime-local

- When the date and time is defined as the type of the input control, a date time picker is provided on a form to select the date, month, year, and time.
- You can select the date and time according to your appropriate time zone.
- In addition, you can set the minimum and maximum date and time values by using the min and max attributes.
- If you want to convert the seconds into the milliseconds, you can use the step attribute.

123

Input types : datetime-local example

```
<!DOCTYPE html>
<html>
<body>
<h2>Local Date Field</h2>
<p>The <strong>input type="datetime-local"</strong>
  specifies a date and time input field, with no time
  zone.</p>
<form action="/action_page.php">
  <label for="birthdaytime">Birthday (date and
  time):</label>
  <input type="datetime-local" id="birthdaytime"
  name="birthdaytime“>
```

124

Input types : datetime-local example

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

<p><strong>Note:</strong> type="datetime-local" is not
   supported in Internet Explorer 11 or prior Safari
   14.1.</p>

</body>
</html>
```

125

Input types : datetime,date,month,week and time

- You can also set the type of the input control to datetime, date, month, week, and time separately.
- The datetime type allows you to enter year, month, day, hour, minute, second, and fraction of a second in the Coordinated Universal Time (UTC).
- The date type allows you to enter the day, month, and year by using the date time picker.
- The month type allows you to enter the month in the input control. You can also set the minimum and maximum month by specifying the min and max attributes of the month type, respectively.
- The week type is used to enter a week of a month.
- The time type is used to enter only time in the input control.

126

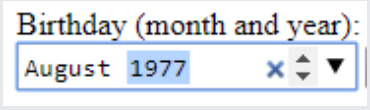
Input types : month

- Allows the user to select a month and year.
- Depending on browser support, a date picker can show up in the input field.
- Syntax : <input type="month">

Example

```
Birthday (month and year):
<input type="month" name="bdaymonth">
```

Result



127

Input types : month example

```
<!DOCTYPE html>
<html>
<body>
<h2>Month Field</h2>
<p>The <strong>input type="month"</strong> allows
   the user to select a month and year.</p>
<form action="/action_page.php">
  <label for="bdaymonth">Birthday (month and
   year):</label>
  <input type="month" id="bdaymonth"
   name="bdaymonth“>
```

128

Input types : month example

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

<p><strong>Note:</strong> type="month" is not
   supported in Firefox, Safari, or Internet Explorer
   11.</p>

</body>
</html>
```

129

Input types : week

- Allows the user to select a week and year.
- Depending on browser support, a date picker can show up in the input field.
- Syntax : <input type="week">

Example

```
Select a week and year:
<input type="week" name="week_year">
```

Result

130

Input types : week example

```
<!DOCTYPE html>
<html>
<body>
<h1>Display a Week Input Control</h1>
<p>The <strong>input type="week"</strong> allows the
   user to select a week and year.</p>
<p>If the browser supports it, a date picker pops up when
   entering the input field.</p>
<form action="/action_page.php">
  <label for="week">Select a week:</label>
  <input type="week" id="week" name="week“>
```

131

Input types : week example

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

<p><strong>Note:</strong> type="week" is not
   supported in Firefox, Safari or Internet Explorer
   11.</p>

</body>
</html>
```

132

Input types : email

- Used for input fields that should contain an e-mail address. The email type of the input control is used to enter a valid e-mail address, which should contain a @ symbol and a dot (.).
- Depending on browser support, the e-mail address can be automatically validated when submitted.
- Syntax : <input type="email">
- You can also enter more than one e-mail address in the e-mail field by using the multiple attribute, which specifies that the multiple e-mail addresses can be entered in the input control.

133

Input types : email

Example

```
E-mail:
<input type="email" name="mail">
```

Result

E-mail:

134

Input types : email example

```
<!DOCTYPE html>
<html>
<body>
<h2>Email Field</h2>
<p>The <strong>input type="email"</strong> is used for
   input fields that should contain an e-mail address:</p>
<form action="/action_page.php">
  <label for="email">Enter your email:</label>
  <input type="email" id="email" name="email">
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

135

Input types : number

- Defines a numeric input field.
- You can also set restrictions on what numbers are accepted by using min and max attributes.
- Syntax : <input type="number">

Example

```
Enter Your Office No. (between 101 and 112):
<input type="number" name="quantity" min="101" max="112">
```

Result

Enter Your Office No. (between 101 and 112): 107

136

Input types : number example

```
<!DOCTYPE html>
<html>
<body>
<h2>Number Field</h2>
<p>The <strong>input type="number"</strong> defines
   a numeric input field.</p>
<p>You can use the min and max attributes to add
   numeric restrictions in the input field:</p>
<form action="/action_page.php">
  <label for="quantity">Quantity (between 1 and
   5):</label>
```

137

Input types : number example

```
<input type="number" id="quantity" name="quantity"
   min="1" max="5">
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

138

Input attribute : step

- The input step attribute specifies the legal number intervals for an input field.
- Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.
- Tip: This attribute can be used together with the max and min attributes to create a range of legal values.
- The step attribute works with the following input types: number, range, date, datetime-local, month, time and week.

139

Input types : number step example

```
<!DOCTYPE html>
<html>
<body>
<h2>Numeric Steps</h2>
<p>Depending on browser support: Fixed steps will
  apply in the input field.</p>
<form action="/action_page.php">
  <label for="quantity">Quantity:</label>
  <input type="number" id="quantity" name="quantity"
   min="0" max="100" step="10" value="30">
  <input type="submit" value="Submit">
```

140

Input types : number step example

```
</form>

</body>
</html>
```

141

Input types : number restrictions

Attribute	Description
checked	Specifies that an input field should be pre-selected when the page loads (for type="checkbox" or type="radio")
disabled	Specifies that an input field should be disabled
max	Specifies the maximum value for an input field
maxlength	Specifies the maximum number of character for an input field
min	Specifies the minimum value for an input field
pattern	Specifies a regular expression to check the input value against
readonly	Specifies that an input field is read only (cannot be changed)
required	Specifies that an input field is required (must be filled out)

142

Input types : number restrictions

Attribute	Description
size	Specifies the width (in characters) of an input field
step	Specifies the legal number intervals for an input field
value	Specifies the default value for an input field

143

Input attribute : pattern

- The input pattern attribute specifies a regular expression that the input field's value is checked against, when the form is submitted.
- The pattern attribute works with the following input types: text, date, search, url, tel, email, and password.

144

Input attribute : pattern example

```
<!DOCTYPE html>
<html>
<body>
<h1>The input pattern attribute</h1>
<p>The pattern attribute specifies a regular expression that the input element's value is checked against.</p>
<form action="/action_page.php">
  <label for="country_code">Country code:</label>
  <input type="text" id="country_code" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code"><br><br>
```

145

Input attribute : pattern example

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

<p><strong>Note:</strong> The pattern attribute of the input tag is not supported in Safari 10 (or earlier).</p>

</body>
</html>
```

146

Input attribute : placeholder

- The input placeholder attribute specifies a short hint that describes the expected value of an input field (a sample value or a short description of the expected format).
- The short hint is displayed in the input field before the user enters a value.
- The placeholder attribute works with the following input types: text, search, url, tel, email, and password.

147

Input attribute : placeholder example

```
<!DOCTYPE html>
<html>
<body>
<h1>The input placeholder attribute</h1>
<p>The placeholder attribute specifies a short hint that describes the expected value of an input field.</p>

<form action="/action_page.php">
  <label for="phone">Enter a phone number:</label>
  <input type="tel" id="phone" name="phone" placeholder="123-45-678" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}"><br><br>
  <input type="submit" value="Submit">
</form>      </body>      </html>
```

148

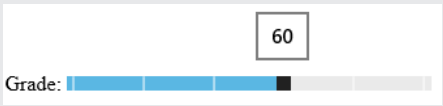
Input type : range

- Defines a control for entering a number whose exact value is not important.
- Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min max and step attributes.
- Depending on browser support, the input type "range" can be displayed as a slider control.
- Syntax <input type="range">

Example

```
Grade:
<input type="range" name="points" min="0" max="100">
```

Result



9

Input type : range

```
<!DOCTYPE html>
<html>
<body>
<h2>Range Field</h2>
<p>Depending on browser support: The input type
"range" can be displayed as a slider control.</p>
<form action="/action_page.php" method="get">
  <label for="vol">Volume (between 0 and 50):</label>
  <input type="range" id="vol" name="vol" min="0"
max="50">
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

150

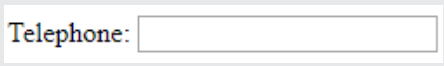
Input type : tel

- Used for input fields that should contain a telephone number.
- The tel type is currently supported only in Safari 8.
- Syntax <input type="tel">

Example

```
Telephone:
<input type="tel" name="telephone">
```

Result




```
<form>
 <label for="phone">Enter your phone number:</label>
 <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">  </form>
```

151

Input type : tel

- In this type, you can only enter numbers, as it does not accept alphabets.
- There is no specific syntax to specify the telephone number in the input control, as there is a wide variety of valid telephone numbers.
- You can also specify particular pattern to enter the telephone number by using the pattern attribute of the INPUT tag.

152

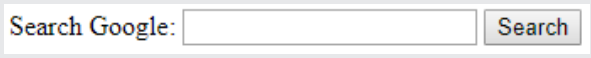
Input type : search

- Used for search fields.
- A search field behaves like a regular text field.
- Syntax <input type="search">

Example

```
Search Google:
  <input type="search" name="googlesearch">
  <input type="submit" value="Search">
```

Result



153

Input type : search

```
<!DOCTYPE html>
<html>
<body>
<h2>Search Field</h2>
<p>The <strong>input type="search"</strong> is used
for search fields (behaves like a regular text field):</p>
<form action="/action_page.php">
  <label for="gsearch">Search Google:</label>
  <input type="search" id="gsearch" name="gsearch">
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

154

Input type : url

- Used for input fields that should contain a URL address.
- A valid url of a website consists of a protocol, a domain name, and a pathname. The most widely used protocols are Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP).
- Depending on browser support, the url field can be automatically validated when submitted.
- Syntax <input type="url">

Example

```
Add your homepage:
  <input type="url" name="homepage">
  <input type="submit" value="Submit">
```

Result

Add your homepage: Submit

155

Input type : file

- The <input type="file"> defines a file-select field and a "Browse" button for file uploads.
- The file type allows a user to select a file stored in the local computer and send it to the server on submission on the form.
- A user can either type the pathname of the file directly into the file selection field or use the browse option to select the pathname of the file from a system-specific dialog box.
- In case of using the browse option, even if the pathname exceeds the maxlength specified, the browser accepts the complete pathname.

156

Input type : file

- The attributes of the file selection field are as follows :
  - **Size** - Defines the width of the visible text on the file selection field
  - **Maxlength** - Specifies the maximum number of character that can be entered in the file selection field
  - **Accept** - Specifies the type of files that can be submitted through a file upload

157

Input type : file

- Example

```
<!DOCTYPE html>
<html>
<body>
<h1>File upload</h1>
<p>Show a file-select field which allows a file to be
chosen for upload:</p>
<form action="/action_page.php">
  <label for="myfile">Select a file:</label>
  <input type="file" id="myfile" name="myfile">
<br><br>   <input type="submit" value="Submit">
</form>
</body>
</html>
```

158

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.
- Example

Input type : image

```
<!DOCTYPE html>
<html>
<body>
<h2>Display an Image as the Submit button</h2>
<form action="/action_page.php">
  <label for="fname">First name: </label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name: </label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="image" src="button.png" alt="Submit" width="48"
height="48">
</form>
<p><b>Note:</b> The input type="image" sends the X and Y
coordinates of the click that activated the image button.</p>
</body>
</html>
```

Submitting a form

- When user clicks the submit button of the form the information contained in the form is send to the server for processing.
- The whole process of transmission of the information from browser to the server is performed by the **enctype, action,** and **method** attributes of the FORM tag.
- The **enctype** attribute encodes the information from the form before transmitting it from the browser to the server.
- The **action** attribute specifies the URL of the program that handles the information contained in the form and the **method** attribute specifies how the information is transmitted from the browser to the server.

161

Enctype attribute

- The enctype attribute specifies how the form-data should be encoded when submitting it to the server, so that the information is not corrupted during the transmission. The encoding is performed by the browser.
- **Note:** The enctype attribute can be used only if method="post".
- Syntax : <form enctype="value">

Value	Description
application/x-www-form-urlencoded	Default. All characters are encoded before sent (spaces are converted to "+" symbols, and special characters are converted to ASCII HEX values)
multipart/form-data	This value is necessary if the user will upload a file through the form
text/plain	Sends data without any encoding at all. Not recommended

162

Enctype attribute

- Three types of encoding process are application/x-www-form-urlencoded, multipart/form-data, and text/plain.
- The application/x-www-form-urlencoded encoding is the standard form of encoding. If no value is set for the enctype attribute, browser accepts the standard type.
- The multipart/form-data encoding is required for those forms that contain file selection field.
- You can use the text/plain attribute along with the mailto URL to send the form information to an e-mail address rather than a server.

163

The application/x-www-form-urlencoded Encoding

- The application/x-www-form-urlencoded encoding acts as a default type of encoding for the browser.
- This implies that if the enctype attribute is not set with any value, the browser by default uses the application/x-www-form-urlencoded encoding.
- In this encoding, the browser converts a space into a a plus sign (+), a non-alphanumeric character into a percent sign (%) followed by two hexadecimal digits that are ASCII code of character, and line breaks into CR/LF pairs (%0D%0A).
- The standard encoding also contains the name of each form controls that is present in the form.
- Each control name is separated from other control name by an ampersand (&).

164

The application/x-www-form-urlencoded Encoding

- Example :
- Let's consider a form, which contains four controls named username, password, name, and address. The values of these controls are richjones, rich23, richard, and Sydney, Australia respectively.
- In such a case, the encoding of the information in the form is done in the following way :

Username=richjones&password=rich23&name=richard&address=Sydney,+Australia

165

The multipart/form-data Encoding

- The multipart/form-data encoding is used only when the method attribute of the form is set to post.
- In this type of encoding, the controls in the form are segregated in several parts.
- This encoding is more cumbersome and longer than the application/x-www-form-urlencoded encoding.
- **Example :**
- Let's consider the example used in the application/x-www-form-urlencoded encoding.
- The form contains four controls named username, password, name, and address, and their values are richjones, rich23, richard, and Sydney, Australia, respectively. The encoding of the given example in the multipart/form-data encoding is as follows:

166

The multipart/form-data Encoding

```
-----------------------------------------------------------1234567897987
  Content-Disposition: form-data; name="username"
    richjones
-----------------------------------------------------------1234567897987
  Content-Disposition: form-data; name="password"
    rich23
-----------------------------------------------------------1234567897987
  Content-Disposition: form-data; name="name"
    richard
-----------------------------------------------------------1234567897987
  Content-Disposition: form-data; name="address"
     Sydney, Australia
-----------------------------------------------------------1234567897987
```

167

The multipart/form-data Encoding

- This type is preferred only when the form contains one or more file selection control.
- Let's consider an example where the form contains a file selection control named **thefile** and the filename is **file**.
- In this case, the multipart/form-data encoding is given as follows:

```
• -----------------------------------------------------------1234567897987
  Content-Disposition: form-data; name="thefile"; filename="file"

    Content of the file...
    -----------------------------------------------------------
  1234567897987
```

The text/plain Encoding

- The text/plain encoding is used when the browser does not have access to the server that contains the form-handling program.
- In this case, the information form the form is sent through an e-mail and the action attribute is set to a mailto URL.
- In this type of encoding, each control is written in a single line, with name and value separated by an equal to sign (=).
- Space is denoted by space and line break is denoted by CR/LF pairs (%0D%0A).

169

The text/plain Encoding

- Example of text/plain Encoding
- Let's again consider the example used in the application/x-www-form-urlencoded encoding in which, there are four controls named username, password, name, and address.
- The encoding of the information in the form is given as follows:

```
username=richjones
password=rich23
name=richard
address=Sydney, Australia
```

170

Example

```
<!DOCTYPE HTML>
	<HEAD>
		<TITLE>Using the enctype, action, and method
Attributes </TITLE>
	</HEAD>
	<BODY>
		<FORM action="action.html" method="get"
enctype="text/plain">
			First name: <INPUT type="text"
name="fname" /><BR />
			Last name: <INPUT type="text"
name="lname" /><BR />
			<INPUT type="submit" value="Submit" />
		</FORM>
	</BODY>
</HTML>
```

171

action.html

```
<!DOCTYPE HTML>
	<HEAD>
		<TITLE>Form Submitted</TITLE>
	</HEAD>
	<BODY>
		<H1> The form is successfully submitted. </H1>
	</BODY>
</HTML>
```

172

Example

```
<!DOCTYPE html>
<html>
<body>

<h1>The form enctype attribute</h1>

<form action="/action_page_binary.asp" method="post"
enctype="multipart/form-data">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

173

Example

```
<form action="/tutorial/action" method="post"
enctype="multipart/form-data"
     onsubmit="alert('Form data has been posted to the
server.');return false;">

  <label for="firstname">First name </label> <br />
  <input type="text" id="firstname" name="firstname"> <br />

  <label for="lastname">Last name </label> <br />
  <input type="text" id="lastname" name="lastname"> <br />

  <label for="avatar">Avatar </label> <br />
  <input type="file" id="avatar" name="avatar"
        accept="image/*"><br /><br />

  <button type="submit">Submit</button>
</form>
```

174


THANKS