



NextGen



Web



Session: 10

HTML Forms



Objectives

- Describe HTML5 forms
- Explain the working of new input types in HTML5
- Explain the new Form attributes
- Explain the new Form elements



New Features in HTML5 Forms

The following are the changes introduced in HTML5 forms:

New form elements

New input types

New attributes

Browser-based validation

CSS3 styling techniques

Forms API

New Elements & Input Type in HTML5

Elements
<code>progress</code>
<code>meter</code>
<code>datalist</code>
<code>output</code>

Input Type
<code>email</code>
<code>search</code>
<code>url</code>
<code>tel</code>
<code>number</code>

<code>range</code>
<code>date</code>
<code>Week</code>
<code>month</code>
<code>time</code>
<code>datetime</code>
<code>datetime-local</code>
<code>color</code>

New Attribute
<code>placeholder</code>
<code>required</code>
<code>multiple</code>
<code>autofocus</code>
<code>pattern</code>
<code>form</code>



Browser-based Validation

- HTML4 supported the use of custom JavaScript or libraries to perform validation on the client-side browsers.
- These validations ensure that the input fields are checked before the form is submitted to the server for further processing.
- The new attributes in HTML5, such as required and pattern can be used with the input elements to perform validation.
- This relieves the Web developers from writing the custom JavaScript code for performing client-side validation on the Web pages.
- HTML5 also provides advanced validation techniques that can be used with JavaScript to set custom validation rules and messages for the input elements.



CSS Styling Techniques

- can enhance the form elements with the pseudo-class selectors, such as **:required**, **:valid**, and **:invalid**.

```
<style>
  input:required {
    outline: 1px red solid;
    color: green ;
  }
  input:required:valid {
    background-size:10px 10px;
    background-position: right top;
    background-repeat: no-repeat;
  }
  input:required:invalid{
    background-size:10px 10px;
    background-position: right top;
    background-repeat: no-repeat;
    color: red ;
  }
</style>
</head>
<body>
  <form>
    Name: <input type="text" name="name" required="true" /><br/>
    Email: <input type="email" name="emailid" required="true" />
    <input type="submit" value="submit" />
  </form>
```

Name:

Email:

Name:

Email:



Forms API

- HTML5 has introduced JavaScript API for forms to customize validations and processing performed on the forms.
- The new Forms API provides new methods, events, and properties to perform complex validations combining fields or calculations.

Events and Methods	Description
setCustomValidity(message)	Sets the custom error message that is displayed when the form is submitted by the user
checkValidity()	Checks the validity of data entered by the user
oninvalid	Allows script to run only when the element is invalid
onforminput	Allows script to run when a form gets a input from the user
onformchange	Allows script to run when the form changes



Working with New Input Types

- The **type** attribute of the **input** element determines what kind of input will be displayed on the user's browser.
- The default input type is "text".
- the list of input types:
 - text
 - label
 - radio
 - textarea
 - checkbox
 - submit
 - email
 - url
 - tel
 - number
 - range
 - Datetime:time,date,week,month,datetime-local
 - color



New Form Attributes

required
placeholder
Pattern:"[0-9]{6,20}"
multiple
autofocus

- HTML5 has provided several new attributes that perform the validations without writing JavaScript snippets.
- They perform the following tasks:
 - Check data provided by users with the regular expression pattern assigned to the fields
 - Inform users with appropriate errors
 - Check the required fields are not left empty
 - Enable multiple values for the fields

Form

- Earlier, all the form controls need to be provided between the opening and closing `<form>` tag.
- In HTML5, elements can be inserted at any place in the document and they can reference the form using the `form` attribute.

```
<body>
    <input type="text" name="mytext" id="mytext"
        form="myform" />
    . . .

    <form id="myform" >
        . . .
    </form>
</body>
```



Autocomplete Attribute

- Input elements that can support autocomplete are text, url, tel, password, datepickers, range, and color.
- The autocomplete feature comprises two states: on and off.
 - on : unsensitive data can be remembered by the browser.
 - off : sensitive data will not be remembered because it is unsafe for storing.
- By default, many browsers have the autocomplete feature enabled.
- The browsers that do not support autocompletion, can be turned on or off for this behavior by autocomplete attribute.

```
E-mail: <input type="email" name="email" autocomplete="off" />  
        <input type="submit" value="submit" />
```

E-mail:

john.smith@gmail.com



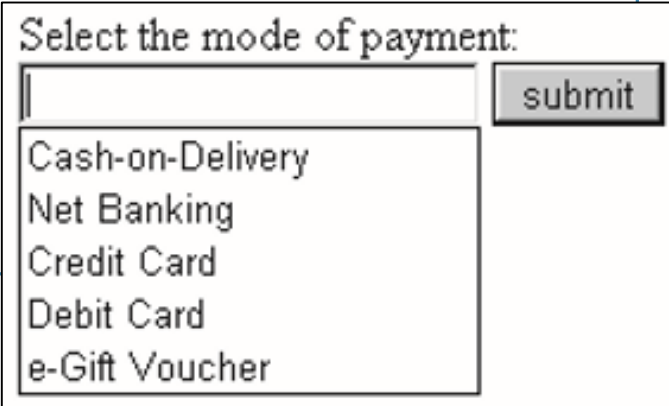
New Form Elements

Datalist
Progress
Meter
Output

- The new form elements are specifically designed to be used with the JavaScript for more functional.
- At present, all the browsers do not provide the support for these new elements.
- If the control is not supported by the browser, then it displays element as a text field.
- Opera provides the support for all the new form elements.

- At present, only Opera browser provides the support for the datalist.

```
<label> Select the mode of payment: </label>
<input type="text" name="payment" list="paymentlist" />
<datalist id="paymentlist" >
  <option value="Cash-on-Delivery" >
  <option value="Net Banking" >
  <option value="Credit Card" >
  <option value="Debit Card" >
  <option value="e-Gift Voucher" >
</datalist>
<input type="submit" value="submit" />
```



The screenshot shows a web form with the label "Select the mode of payment:". Below the label is a text input field that has expanded into a dropdown menu. The dropdown menu lists five options: "Cash-on-Delivery", "Net Banking", "Credit Card", "Debit Card", and "e-Gift Voucher". To the right of the input field is a "submit" button.



Progress

- Represents the current status of a task, which gradually changes as the task heads for completion.
- This is not a form-specific element.
- For example, when the user downloads any file from a particular Web page, the download task is represented as a progress bar.

```
<label> Downloading status: </label>  
<progress value="35" max="100" > </progress>  
<input type="submit" value="submit" />
```

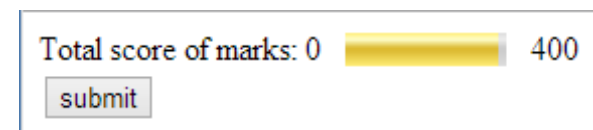
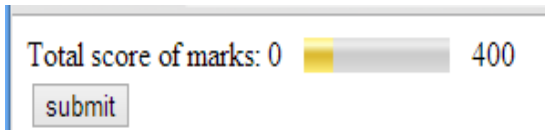
Downloading status: 



Meter

- Represents a measurement scale for a known range.
- Has a definite minimum and maximum values
- For example, a meter element can be used to represent measurements, such as disk usage space, fraction value, or significance of a query result.
- All these have a known maximum value defined for them.

```
<label> Total score of marks: </label>
0 &nbsp; <meter min="0" max="400" value="350"
    title="numbers scored" low="120" high="300">
    </meter> &nbsp; 400<br/>
<input type="submit" value="submit"/>
```





Output

Membership Type:	<input type="text" value="Silver - \$500"/>
Duration [years]:	<input type="text" value="3"/>
Annual Payment Fees: \$. 1500	

- Displays the results of a calculation on a form.
- The result values are processed from the other form elements.

```
<form oninput="x.value = parseInt(type.value) *  
                parseInt(duration.value)" >  
  
  <label> Membership Type: </label>  
  <select name="type" >  
    <option value="400" > Gold - $400 </option>  
    <option value="500" > Silver - $500 </option>  
    <option value="600" > Platinum - $600 </option>  
  </select>  
  
  <label> Duration [years]: </label>  
  <input type="number" value="0" name="duration"  
        min="1" max="5" step="1" />  
  
  <label> Annual Payment Fees: $. </label>  
  
  <output name="x" for="type duration"> </output>
```




Summary

- HTML5 provides a great enhancement to Web forms.
- Creation of form is made easier for Web developers by standardizing them with rich form controls.
- HTML5 introduces new form elements such as new input types, new attributes, browser-based validation, CSS3 styling techniques, and forms API.
- HTML5 provides new input types that are data-specific user interface elements such as email, url, number, range, date, tel, and color.
- The new form elements introduced in HTML5 are datalist, progress, meter, and output.
- HTML5 has provided several new attributes that performs the validations without writing JavaScript snippets for them.
- In HTML5, one can use the submit input type for form submission.