MODULE 1: PERFORM PROGRAMMING IN HTML5 WITH JAVASCRIPT AND CSS3

Week 4-5

UNIT 2: Create HTML5 document using advanced techniques with JavaScript and CSS3

Most Essential Learning Competencies: *At the end of the course, you must be able to:*

1. Discuss web forms and its attributes in html5 using input types
2. Create sample form with input types

Reading Activity

Forms are best learnt using a hands on approach. To become proficient with HTML forms you need to create many, sorting out the problematic nuances as you go along. Therefore, the main content of the unit is a series of sections: the first is a short introduction to HTML forms; the second discusses each form element, and involves some textbook study. (You may find it more convenient to postpone activities until you have covered all the form elements).

This introduction covers the main form elements. It also explains the process that occurs when a form is submitted. The main elements of forms are: Text fields; Password fields; Text areas; Radio buttons; Check boxes; Menu buttons and scrolling lists; Submit and reset buttons; and file picker. HTML5 defines a number of new input types that can be used in forms. Examples are Email address fields; web address fields; numbers as spin boxes and sliders; date pickers; search boxes; color pickers; form validation; and required fields. We will look at some of these in this chapter.

Processing Forms

Although forms could simply be used to display information, HTML provides them in order to supply a way for the user to interact with a Web server. The most widely used method to process the data submitted through form is to send it to server-side software typically written in a scripting language, although any programming language can be used. The figure below outlines the kind of processing that takes place.

CP121

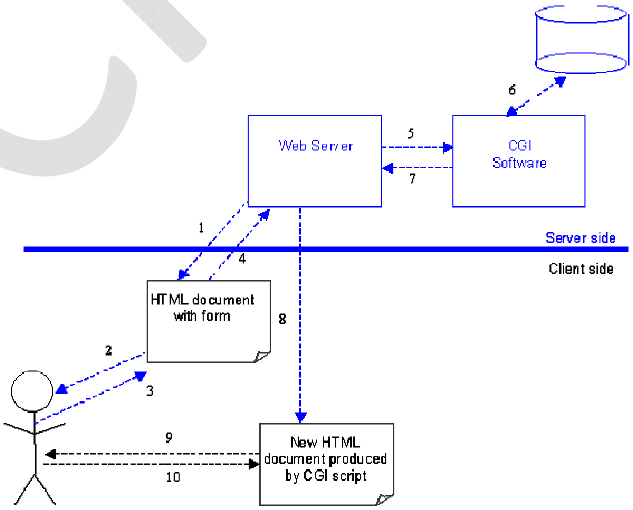

Figure 4-5

1. The user retrieves a document containing a form from a Webserver.
2. The user reads the Web page and interacts with the form itcontains.
3. Submitting the form sends the form data to the server for processing.
4. The Web server passes the data to a CGI program.
5. The CGI software may use database information or store data in a server-side database.
6. The CGI software may generate a new Web page for the server to return to the user.
7. The user reads the new Web document and may interact with it.

Prior to HTML5, developers and web designers had to write a lot of code for basic things such as a datepicker. JavaScript was used extensively for basic things and it was difficult to write such code given the complexity of the web pages we come across nowadays. HTML5 accounts for cleaner code and the introduction of new input types makes it easier for developers to design their web pages. In this chapter, we will discuss the new input types and attributes in HTML5. If the input types and attributes are not compatible with earlier versions of the browsers, they will be ignored and the <input type= text> default will be considered.

Currently, some of the features of HTML5 are supported by the latest versions of Opera and Google Chrome. However, certain features of HTML5 are not supported by these browsers. Firefox support is still in the pipeline. After HTML5 is accepted as a norm, these features will be standardized and will be supported by all browsers.

New Form Elements

Tag	Description
<datalist>	Defines pre-defined options for input controls
<keygen>	Defines a key-pair generator field (for forms)
<output>	Defines the result of a calculation

New Input Types

New Input Types	New Input Attributes
• **Color**	• autocomplete
• **date**	• autofocus
• **datetime**	• form
• **datetime-local**	• formaction
• **email**	• formenctype
• **month**	• formmethod
• **number**	• formnovalidate

- **range**
- **search**
- **tel**
- **time**
- **url**
- **week**

- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

HTML5 < datalist> Element

The <datalist> tag specifies a list of pre-defined options for an <input> element.

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

Use the <input> element's list attribute to bind it together with a <datalist> element.

HTML5 < keygen> Element

The <keygen> tag specifies a key-pair generator field used for forms.

When the form is submitted, the private key is stored locally, and the public key is sent to the server.

HTML5 < output> Element

The <output> tag represents the result of a calculation.

The <output> element is the semantically correct element for displaying the results of a calculation from form elements.

- Attributes include for, form and name, in addition to the global attributes.
- It does not have the value attribute. Rather, the textnode between the opening and closing tags make up the value. The default being the empty string.
- It really doesn't need the value attribute, since the name/value pair are NOT submitted along with the form
- For the most part, it's just a simple semantic element that any browser supports in display but not necessarily in function: similar to a <span> in appearance, but with added functionality.
- The for attribute takes a space-separated (not comma-separated) list of ID's of elements that went into calculation.
- The form attribute associates the <output> with a form, which is useful if it is not

CP1121

nested within it.

- If the form is reset, the value will return to its default value.
- Example usage: 1) subtotal, tax and total in a shopping cart, 2) output ofa calculator, 3) current value of a range input.

New Input Types

New	Description
color	Defines a color picker
date	Defines a date control (year, month and day (no time))
datetime	The input type datetime has been removed from the HTML standard. Use datetime-local instead.
datetime-local	Defines a date and time control (year, month, day, hour, minute, second, and fraction of a second (no time zone)
email	Defines a field for an e-mail address
file	Defines a file-select field and a "Browse..." button (for file uploads)
hidden	Defines a hidden input field
image	Defines an image as the submit button
month	Defines a month and year control (no time zone)
number	Defines a field for entering a number
range	Defines a control for entering a number whose exact value is not important (like a slider control)
reset	Defines a reset button (resets all form values to default values)
search	Defines a text field for entering a search string
tel	Defines a field for entering a telephone number
time	Defines a control for entering a time (no time zone)
url	Defines a field for entering a URL
week	Defines a week and year control (no time zone)

Input type: color

Select a color from a color picker:

Input type: date

Define a date control:

Input type: datetime

Define a date and time control (with time zone):

The input type datetime has been removed from the HTML standard. Use datetime-local instead.

Input type: datetime-local

Define a date and time control (no time zone):

Input type: email

Define a field for an e-mail address (will be automatically validated when submitted):

Input type: file

Define a file-select field and a "Browse..." button (for file uploads):

Input type: image

Define an image as a submit button:

Input type: month

Define a month and year control (no time zone):

Input type: number

Define a field for entering a number (You can also set restrictions on what numbers are accepted):

Use the following attributes to specify restrictions:

- max - specifies the maximum value allowed
- min - specifies the minimum value allowed
- step - specifies the legal number intervals
- value - Specifies the default value

Input type: range

Define a control for entering a number whose exact value is not important (like a slider control). You can also set restrictions on what numbers are accepted:

Use the following attributes to specify restrictions:

- max - specifies the maximum value allowed
- min - specifies the minimum value allowed
- step - specifies the legal number intervals
- value - Specifies the default value

Input type: reset

Define a reset button (resets all form values to default values):

CP121

Input type: search
Define a search field (like a site search, or Google search):

Input type: tel
Define a field for entering a telephone number:

Input type: time
Define a control for entering a time (no time zone):

Input type: url
Define a field for entering a URL:

Input type: week
Define a week and year control (no time zone):

New Form Attribute

The placeholder attribute

HTML5 introduced a new attribute called **placeholder**. This attribute on <input> and <textarea> elements provides a hint to the user of what can be entered in the field. The place holder text must not contain carriage returns or line-feeds.

The required attribute

Now you do not need to have javascript for client side validations like empty text box would never be submitted because HTML5 introduced a new attribute called **required** which would be used as follows and would insist to have a value

The multiple Attribute

The multiple attribute is a Boolean attribute. When present, it specifies that the user is allowed to enter more than one value in the <input> element. The multiple attribute works with the following input types: email, and file.

The step Attribute

The step attribute specifies the legal number intervals for an <input> element. Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

Tip: The step 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, datetime-local, month, time and week.

CP121

Self-Check

Quiz 4-5.1

Instructions: Write your answer on the Answer Sheet (AS) provided in this module (1-point each).

1-7. Enumerate the processing of web forms
8-10. New web form elements tags

Laboratory Activity

Activity 4-5.1

Adding Web Forms in HTML5

A. Open **Brackets** (code for the web) text editor or any text editor of your choice.

B. Copy-paste each html codes below and save all files respectively the folder named **HTML5 Web Forms** in your desktop and open the file using your available web browser to see the output.

1. **Filename: datalist.html**

```
<!DOCTYPE html>
<html>
<body>
<form >
<input list= "browsers" name="browser">
<datalist id= "browsers" >
<option value="Internet Explorer">
<option value="Firefox">
<option value="Chrome">
<option value="Opera">
<option value="Safari">
</datalist>
<input type="submit">
</form>
</body>
</html>
```

2. Filename: output.html.

```
<!DOCTYPE html>
<html>
<body>
<form oninput= "x.value=parseInt(a.value) + parseInt(b.value)" >0
<input type ="range" id="a" value="50"> 100 +
<input type="number" id="b" value="50">
= <output name="x" for="a b"></output>
</form>
</body>
</html>
```

3. Filename: **calendar.html.**

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Birthday: <input type="date" name="bday">
<input type="submit">
</form>
</body>
</html>
```

4. Filename: datetime.html.

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Birthday (date and time): <input type="datetime" name="bdaytime">
<input type="submit">
</form>
</body></html>
```

5. Filename: email.html

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
E-mail: <input type="email" name="usremail">
```

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

6. **Filename: file.html**

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Select a file: <input type="file" name="img">
<input type="submit">
</form>
</body></html>
```

7. **Filename: number.html**

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Quantity (between 1 and 5): <input type="number" name="quantity" min="1" max="5">
<input type="submit">
</form>
</body>
</html>
```

8. **Filename: range.html**

```
<!DOCTYPE html>
<html>
<body>
<form action="#">
Points: <input type="range" name="points" min="0" max="10">
<input type="submit">
</form>
</body>
</html>
```

9. **Filename: placeholder.html**

```
<!DOCTYPE HTML>
<html>
<body>
<form action="#" >
```

```
Enter email : <input type="email" name="newinput"
placeholder="email@example.com"/>
<input type="submit" value="submit" />
</form>
</body>
</html>
```

10. Filename: required.html

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

<form action="/cgi-bin/html5.cgi" method="get">
Enter email : <input type="text" name="newinput" required/>
<p>Try to submit using Submit button</p>
<input type="submit" value="submit" />
</form>

</body>
</html>
```

End of Activity

Read Additional Resources

1. Week4-5_HTML5 Tutorials.pdf

Watch Video Resources

1. Week4-5_The Form Element.mp4

Internet References

1. https://w3schools.com
2. https://www.youtube.com/watch?v=65N6g_Uso4M
3. https://www.tutorialspoint.com/html5/html5_tutorial.pdf