# Working with Forms

Mohamed Mukthar Ahmed

1

# Working with Forms

**Outline**

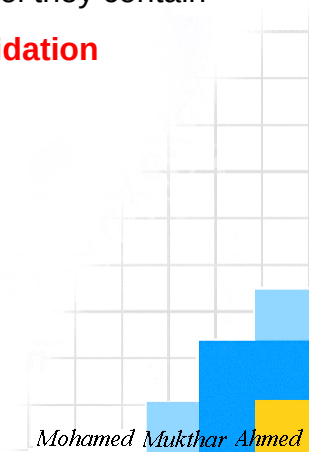
- Using Form Controls
- Form element & Input element
- Attributes of input element
- Using Buttons
- Using Different Types of controls
  - Text fields, Radio Buttons, Checkboxes
  - Drop-down list, List boxes, Textarea, Labels
- Grouping Controls
- File Upload Control
- Aligning Controls
- Tab order and Access key
- HTML5 attributes & CSS3 selectors
- HTML5 controls
- HTML5 Number & Range controls
- Using date & time controls
- Search control and Color control
- Output Element
- Progress & Meter elements

Mohamed Mukthar Ahmed

2

## Working with Forms

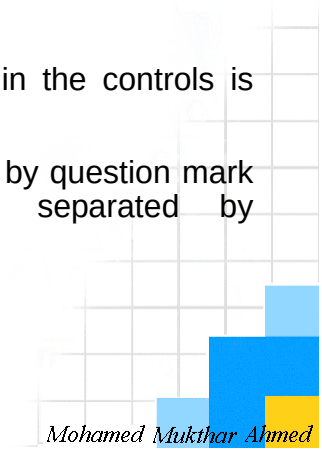
- Forms **let the user enter the data**. The user can click on a button to submit the data on the Web Server for processing
- Let's learn how to code forms and control they contain
- Also learn HTML5 features for **data validation**
- Use **new** HTML5 **controls**




3

## Use Forms & Controls

- A form contains one or more controls like – **text boxes**, **radio buttons**, **lists** or **check boxes**
- We code the **name** attribute to uniquely identify each form and control. We only need to code the **id** attribute if we want to use it as CSS selector
- When a form is submitted, the data in the controls is sent along with the **HTTP request**
- For **GET** method, the URL is followed by question mark and name/value pairs that are separated by ampersands.
- For **POST** method, the data is hidden




4

# Form Element & Input Element

- The **<form>** element defines a form. Its attributes are

| Attribute | Description |
|---|---|
| `name` | A name that can be referred to by client-side or server-side code. |
| `action` | The URL of the file that will process the data in the form. |
| `method` | The HTTP method for submitting the form data. It can be set to either "get" or "post". The default value is "get". |
| `target` | Where to open the page that's specified in the action attribute. If you specify, _blank, the page is opened in a new window or tab. |

- The **<input>** element specifies the controls

| Attribute | Description |
|---|---|
| `type` | The type of control like "button", "text", or "checkbox". |
| `name` | A name that can be referred to by client-side or server-side code. |
| `disabled` | A Boolean attribute that disables and grays out the control. Then, the control can't receive the focus, the user can't tab to it, and the value isn't submitted with the form. |
| `readonly` | A Boolean attribute that means a user can't change the control's value. But the control can receive the focus, the user can tab to it, and the value is submitted with the form. |

Mohamed Mukthar Ahmed

5

# Attributes for input element for btn.

| Attribute | Description |
|---|---|
| `type` | The type of button. Valid values include "submit", "reset", "button", or "image". The "submit" and "image" types submit the form to the server, the "reset" type resets all fields to their default values, and the "button" type is typically used to run a client-side script. |
| `value` | The text that's displayed on the button and submitted to the server when the button is clicked. |
| `src` | For an image button, the relative or absolute URL of the image to display. |
| `alt` | For an image button, alternate text to display in place of the image. |
| `height` | For an image button, the height of the button in either pixels or percent. |
| `width` | For an image button, the width of the button in either pixels or percent. |

Mohamed Mukthar Ahmed

6

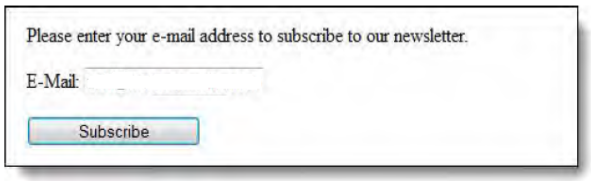
# Form Element & Input Element

- The **<form>** element defines a form. Its attributes are
- The **<input>** element specifies the controls

**The HTML for a form**

```
<form name="email_form" action="subscribe.php" method="post">
    <p>Please enter your e-mail address to subscribe to our newsletter.</p>
    <p>E-Mail: <input type="text" name="email"></p>
    <p><input type="submit" name="submit" value="Subscribe"></p>
</form>
```

**The form in a web browser**




Mohamed Mukthar Ahmed

7

# Using Buttons

- Use **<input>** element to create **FOUR** different types of buttons
- Also use the **<button>** element to create a button
  - The **<input>** element allows a button to contain plain text or image
  - The **<button>** element allows to contain formatted text and other HTML elements like images
- The **submit** button sends form data to the web server as part of HTTP request
- The **reset** button resets all the fields in the form to default value
- Use **button** type to perform processing before the form is submitted to server

Mohamed Mukthar Ahmed

8

# Using Buttons

**Four buttons that are created by the input element**

```
<input type="button" name="message" value="Alert Me">
<input type="submit" name="checkout" value="Checkout">
<input type="reset" name="resetform" value="Reset">
<input type="image" src="images/submit.jpg" alt="Submit button"
       width="114" height="42">
```

**A button that is created by the button element**

```
<button type="submit">
    <img src="images/addtocart.png" width="30" height="23"
        alt="Add to Cart">Add to Cart</button>
```

**The buttons in a web browser**


Mohamed Mukthar Ahmed



# Using Text Fields

- Use **<input>** element to create **THREE** types of text fields
- **Attributes** of input element for text fields

| Attribute | Description |
|---|---|
| `type` | The type of text field. Valid values include "text", "password", and "hidden". |
| `value` | The default value for the field, but the user can change this value. If a reset button is clicked, the field will revert to this value. |
| `maxlength` | The maximum number of characters that the user can enter in the field. |
| `size` | The width of the field in characters based on the average character width of the font. However, it's better to use CSS to set the size of a field. |
| `autofocus` | New to HTML5, a Boolean attribute that tells the browser to set the focus on the field when the page is loaded. |
| `placeholder` | New to HTML5, this attribute puts a default value or hint in the field. Unlike the value attribute, though, this value is removed when the user's cursor enters the control. |

Mohamed Mukthar Ahmed

# Using Text Fields

### The HTML for text fields

```
Quantity:<input type="text" name="quantity" value="1" size="5"
               readonly><br><br>
Username:<input type="text" name="username" autofocus><br><br>
Password:<input type="password" name="password" maxlength="6">
               placeholder="Enter your password"><br><br>
Hidden:<input type="hidden" name="productid" value="widget">
```

### The text fields in a web browser


Mohamed Mukthar Ahmed

11

# Using Radio Buttons and Checkboxes

- Setting the **<input>** element's type attribute to **radio** or **checkbox**
- Only one radio button in a group can be selected at a time. The radio buttons in the group must have the **same name attribute** but different values
- Check boxes are unrelated. More than one check box can be checked at the same time

| Attribute | Description |
|---|---|
| `type` | The type of control, either "radio" or "checkbox". |
| `value` | The value to submit to the server when the control is checked and the form is submitted. |
| `checked` | A Boolean attribute that causes the control to be checked when the page is loaded. If a reset button is clicked, the control reverts to the checked state. |

Mohamed Mukthar Ahmed

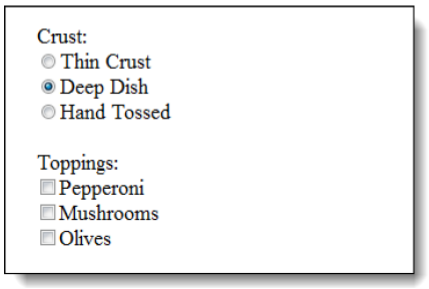
12

# Using Radio Buttons and Checkboxes

**The HTML for radio buttons and check boxes**

```
Crust:<br>
<input type="radio" name="crust" value="thin">Thin Crust<br>
<input type="radio" name="crust" value="deep" checked>Deep Dish<br>
<input type="radio" name="crust" value="hand">Hand Tossed<br><br>
Toppings:<br>
<input type="checkbox" name="topping1" value="pepperoni">Pepperoni<br>
<input type="checkbox" name="topping2" value="mushrooms">Mushrooms<br>
<input type="checkbox" name="topping3" value="olives">Olives
```

**The radio buttons and check boxes in a web browser**






# Using Drop-Down List

- To create a drop-down list use the **<select>** element that contains **<option>** elements
- To group the options in the list, code the options within an **<optgroup>** element
- To use the drop-down list, clicks the arrow at the right side of the field to display the list
- By default the first option is selected in the list when the page is loaded. To change, code the selected attribute for the option

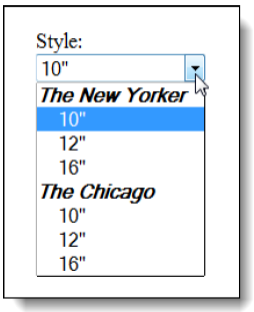
| Element | Attribute | Description |
|---|---|---|
| optgroup | label | The text that's used to identify a group of options. |
| option | value | The value of the selected option that will be sent to the server for processing. |
| option | selected | A Boolean attribute that causes the option to be selected when the page is loaded. |

# Using Drop-Down List

**The HTML for a drop-down list**

```
Style:<br>
<select name="style_and_size">
    <optgroup label="The New Yorker">
        <option value="ny10">10"</option>
        <option value="ny12">12"</option>
        <option value="ny16">16"</option>
    </optgroup>
    <optgroup label="The Chicago">
        <option value="chi10">1
        <option value="chi12">1
        <option value="chi16">1
    </optgroup>
</select>
```

**The drop-down list in a web browser**






# Using List Boxes

- A list box differs from the drop-down list as two or more of its options are always displayed
- The **size** attribute of the **<select>** element specify the number of options that can be displayed

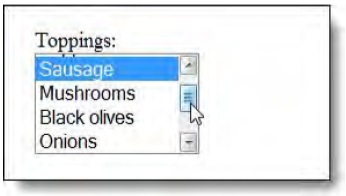
| Attribute | Description |
|---|---|
| `size` | The number of items to display in the control. If the value is 1, the control will be a drop-down list. The default value is 1. |
| `multiple` | A Boolean attribute that determines whether multiple items can be selected. It is only valid if size is greater than 1. |

# Using List Boxes

**The HTML for a list box**

```
<select name="toppings" size="4" multiple>
    <option value="pepperoni">Pepperoni</option>
    <option value="sausage" selected>Sausage</option>
    <option value="mushrooms">Mushrooms</option>
    <option value="olives">Black olives</option>
    <option value="onions">Onions</option>
    <option value="bacon">Canadian bacon</option>
    <option value="pineapple">Pineapple</option>
</select>
```

**The list box in a web browser with a scroll bar**




Mohamed Mukthar Ahmed



# Using Text areas

- Similar to the text field, however, can display multiple lines of text
- Set the **height** and **width** of the **<textarea>** element. We can also use CSS height and width properties

| Attribute | Description |
|---|---|
| `rows` | The approximate number of rows in the text area. Not required in HTML5. |
| `cols` | The approximate number of columns in the text area. Not required in HTML5. |
| `wrap` | Specifies how the text should wrap. Possible values include soft and hard, and soft is the default. |

Mohamed Mukthar Ahmed

# Using Text areas

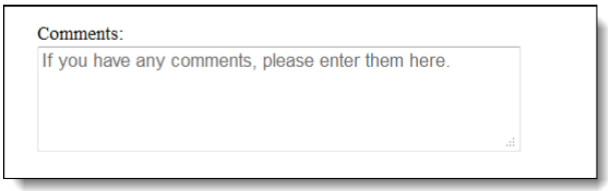
**The HTML for a text area with default text**

```
Comments:<br>
<textarea name="comments"
    placeholder="If you have any comments, please enter them here.">
</textarea>
```

**The CSS for the text area**

```
textarea {
    height: 5em;
    width: 25em;
    font-family: Arial, Helvetica, sans-serif; }
```

**The text area in a web browser**




Mohamed Mukthar Ahmed

19

# Using Labels

- A **label** is commonly used to identify a related field
- Labels should also be **used to improve the readability** of **radio** and **check boxes**
- Labels makes it easier to align the control on the page

| Attribute | Description |
|---|---|
| `for` | Should be set to the id of the related control. Although the id attribute is optional in forms that don't rely on client-side scripting, it is required when using labels and the for attribute. |

Mohamed Mukthar Ahmed

20

# Using Labels

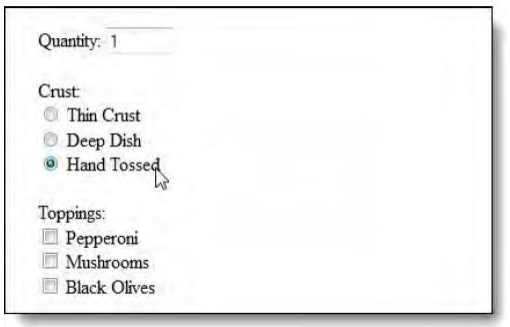
**The HTML for a form with label elements**

```
<label for="quantity">Quantity:</label>
<input type="text" name="quantity" id="quantity" value="1" size="5"><br><br>

Crust:<br>
<input type="radio" name="crust" id="crust1" value="thin">
<label for="crust1">Thin Crust</label><br>
<input type="radio" name="crust" id="crust2" value="deep">
<label for="crust2">Deep Dish</label><br>
<input type="radio" name="crus
<label for="crust3">Hand Tosse

Toppings:<br>
<input type="checkbox" name="t
<label for="topping1">Pepperon
<input type="checkbox" name="t
<label for="topping2">Mushroom
<input type="checkbox" name="t
<label for="topping3">Black Ol
```

**The HTML in a browser**






# Grouping Controls

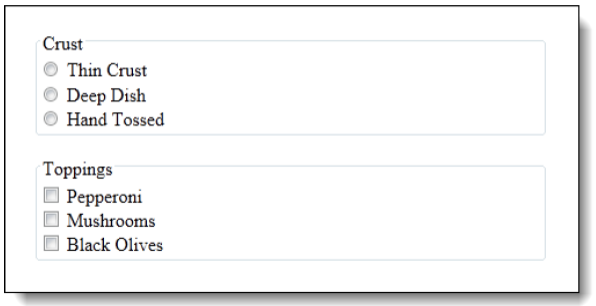
- The **fieldset** element is **used to group controls**
- The **legend** element can be coded within the **fieldset** element. It is **used to label a group of elements**
- If we want to disable all the controls within a **fieldset** element, the **disable** attribute is used

# Grouping Controls

**HTML that uses fieldset and legend elements**

```
<form name="order" action="order.php" method="post">
<fieldset>
    <legend>Crust</legend>
    <input type="radio" name="crust" id="crust1" value="thin">
    <label for="crust1">Thin Crust</label><br>
    <input type="radio" name="crust" id="crust2" value="deep">
    <label for="crust2">Deep Dish</label><br>
    <input type="radio" name="crust" id="crust3" value="hand">
    <label for="crust3"
</fieldset>
<br>
<fieldset>
    <legend>Toppings</l
    <input type="checkb
    <label for="toppinc
    <input type="checkb
    <label for="toppinc
    <input type="checkb
    <label for="toppinc
</fieldset>
</form>
```

**The elements in a web browser**




Mohamed Mukthar Ahmed



# File Upload Control

- To create a file upload control, code the **input** element with "**file**" as type attribute. This allows the user to select the file(s) they want to upload
- In the form element, the method attribute must be set to **POST** and we must code the "**enctype**" attribute.

| Attribute | Description |
|---|---|
| `accept` | The types of files that are accepted for upload. When the operating system's open dialog box opens, only files of those types will be shown. |
| `multiple` | A Boolean attribute that lets the user upload more than one file. |

Mohamed Mukthar Ahmed

# File Upload Control

**The HTML for a file upload element that accepts JPEG images**

```
<form name="upload_form" action="sendemail.php" method="post"
      enctype="multipart/form-data">
    Attach an image:<br>
    <input type="file" name="fileupload" accept="image/jpeg, image/gif">
</form>
```

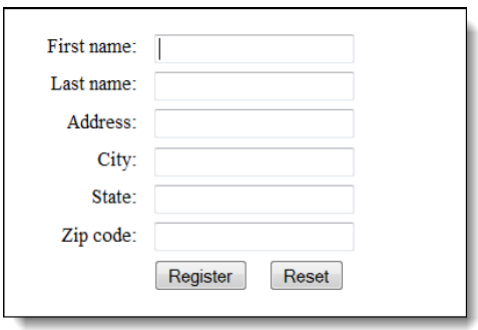
**The file upload control in the Firefox browser**


Mohamed Mukthar Ahmed

25

# Aligning Controls

- If a form includes a series of controls and labels to identify them, we align labels by floating them to left of the controls and set the width that provides enough space for the labels
- Later we set the left-margin for the controls to add space between labels and controls




Mohamed Mukthar Ahmed

26

# Aligning Controls

**The HTML for the form**

```
<label for="firstname">First name:</label>
<input type="text" name="firstname" id="firstname" autofocus><br>
<label for="lastname">Last name:</label>
<input type="text" name="lastname" id="lastname"><br>
<label for="address">Address:</label>
<input type="text" name="address" id="address"><br>
<label for="city">City:</label>
<input type="text" name="city" id="city"><br>
<label for="state">State:</label>
<input type="text" name="state" id="state"><br>
<label for="zip">Zip code:</label>
<input type="text" name="zip" id="zip"><br>
<input type="submit" name="register" id="button" value="Register">
<input type="reset" name="reset" id="reset">
```

**The CSS for the controls**

```
label {
    float: left;
    width: 5em;
    text-align: right;}
input {
    margin-left: 1em;
    margin-bottom: .5em;}
#button {
    margin-left: 7em;}
```



# Tab Order and Access Key

- The **tab order** for a form is the sequence in which the controls receive the focus when **TAB** key is pressed
- **Access keys** are **shortcut keys** that the user can press to move the focus to specific controls on a form
- ***Access key control depends on the browser***
- Setting a proper tab order and providing access keys **improves the accessibility** for users who can't use a mouse

| Attribute | Description |
|---|---|
| `tabindex` | To set the tab order for a control, use a value of 0 or more. To take a control out of the tab order, use a negative value like -1. |
| `accesskey` | A keyboard key that can be pressed in combination with a control key to move the focus to the control. |

# Tab Order and Access Key

### The HTML for the controls

```
<label for="firstname"><u>F</u>irst name:</label>
<input type="text" name="firstname" id="firstname" accesskey="F"><br>
<label for="lastname"><u>L</u>ast name:</label>
<input type="text" name="lastname" id="lastname" accesskey="L"><br>
<label for="email"><u>E</u>mail:</label>
<input type="text" name="email" id="email" accesskey="E">
```

### Another way to define the access keys

```
<label for="firstname" accesskey="F"><u>F</u>irst name:</label>
<input type="text" name="firstname" id="firstname"><br>
<label for="lastname" accesskey="L"><u>L</u>ast name:</label>
<input type="text" name="lastname" id="lastname"><br>
<label for="email" accesskey="E"><u>E</u>mail:</label>
<input type="text" name="email" id="email”>
```

Mohamed Mukthar Ahmed

29

# HTML5 Attributes & CSS3 Selectors

- **HTML5** provides **THREE** attributes for **data validation**

| Attribute | Description |
|---|---|
| `autocomplete` | Set this attribute to off to tell the browser to disable auto-completion. This can be coded for a form or a control. |
| `required` | This Boolean attribute indicates that a value is required for a field. If the form is submitted and the field is empty, the browser displays its default error message. |
| `novalidate` | This Boolean attribute tells the browser that it shouldn't validate the form or control that it is coded for. |

Mohamed Mukthar Ahmed

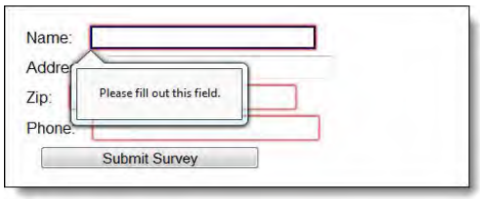
30

# HTML5 Attributes & CSS3 Selectors

**HTML that uses the validation attributes**

```
Name:    <input type="text" name="name" required><br>
Address: <input type="text" name="address" novalidate><br>
Zip:     <input type="text" name="zip" required><br>
Phone:   <input type="text" name="phone" required autocomplete="off"><br>
<input type="submit" name="submit" value="Submit Survey">
```

**The error message and highlighting used by Firefox**




**The CSS3 pseudo-classes for required, valid, and invalid fields**

```
:required
:valid
:invalid
```

**A CSS attribute selector for all controls with the required attribute**

```
input[required]
```

Mohamed Mukthar Ahmed

31

# HTML5 Controls

- **HTML5** provides several **new controls**
- Some of these do validation, some provide better ways to enter data
- We can use **email**, **url** and **tel** controls

| Control | Description |
|---|---|
| `email` | A control for receiving an email address. This implies that the entry will be validated by the browser when the form is submitted. |
| `url` | A control for receiving a URL. This implies that the entry will be validated by the browser when the form is submitted. |
| `tel` | A control for receiving a telephone number, but currently this doesn't imply validation because the formats vary from one country to another. |

Mohamed Mukthar Ahmed

32

# HTML5 Controls

**HTML code that uses the email, url, and tel elements**

```
<form name="email_form" action="survey.php" method="post">
    <h3>Your information:</h3>
    <label for="email">Your email address:</label>
    <input type="email" name="email" id="email" required><br>
    <label for="link">Your web site:</label>
    <input type="url" name="link" id="link"><br>
    <label for="phone">Your phone number:</label>
    <input type="tel" name="phone" id="phone" required><br><br>
    <input type="submit" name="submit" value="Submit Survey">
</form>
```

**The form in Opera**


Mohamed Mukthar Ahmed



# HTML5 Number & Range Controls

- Are designed for **numeric entries**
- Good to use, because they indicate what type of data each control is for
- If these controls are not supported by a browser, text box is displayed

**Attributes for the number and range controls**

| Attribute | Description |
|---|---|
| `min` | The minimum value that may be entered. |
| `max` | The maximum value that may be entered. |
| `step` | The value that the entry is increased or decreased by when the user clicks on the up or down arrow for a number field or moves the slider for a range field. |

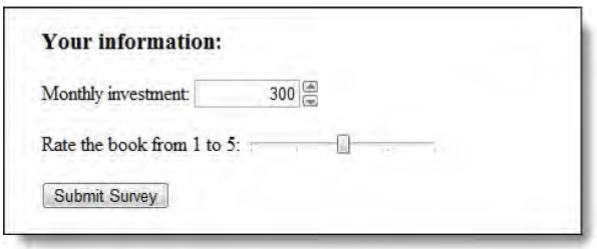
Mohamed Mukthar Ahmed

# HTML5 Number & Range Controls

**HTML that uses number and range controls**

```
<h3>Your information:</h3>
<form name="test_form" action="test.php" method="get">
    <label for="investment">Monthly investment: </label>
    <input type="number" name="investment" id="investment"
           min="100" max="1000" step="100" value="300"><br><br>
    <label for="book">Rate the book from 1 to 5: </label>
    <input type="range" name="book" id="book"
           min="1" max="5" step="1"><br><br>
    <input type="submit" name="submit" value="Submit Survey">
</form>
```

**The form in Opera**




Mohamed Mukthar Ahmed



# Using Data & Time Controls

- **HTML5** **date**, **datetime**, **datetime-local**, **month**, **week** and **time** controls are designed for date and time entries
- Good to **uses** these controls for **semantic reasons**

**Attributes for the date and time controls**

| Attribute | Description |
|---|---|
| `max` | The maximum value that may be entered within a date or time field. |
| `min` | The minimum value that may be entered within a date or time field. |

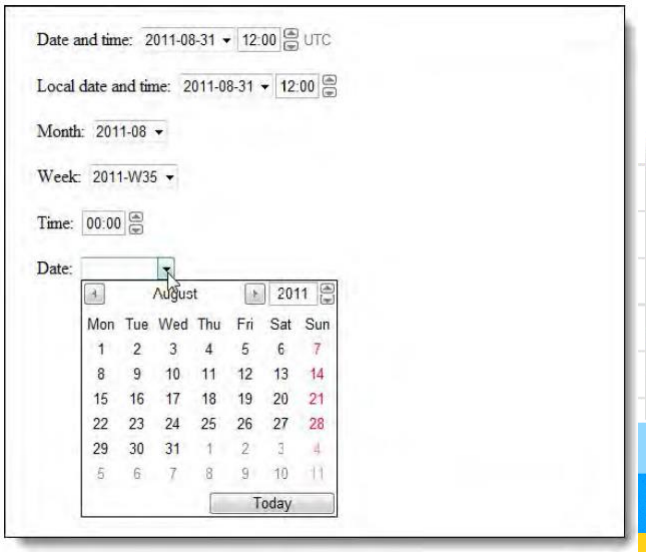
Mohamed Mukthar Ahmed

# Using Data & Time Controls

**HTML that uses th**

```
Date and time:&nb
    <input type=
Local date and t
    <input type=
Month: &nbs
    <input type=
Week:  
    <input type=
Time:  
    <input type=
Date:  
    <input type=
```

**The controls in Opera**



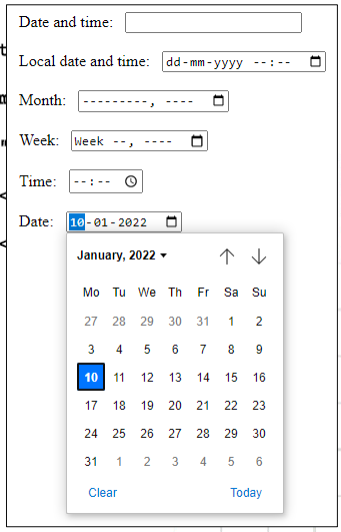



37

# Using Data & Time Controls

**HTML that uses the date and time controls**

```
Date and time:  
    <input type="datetime" name="dat
Local date and time:  
    <input type="datetime-local" nam
Month:  
    <input type="month" name="month"
Week:  
    <input type="week" name="week"><
Time:  
    <input type="time" name="time"><
Date:  
    <input type="date" name="date">
```






38

# Search Control

- To implement a search function, use the **input** element with the "**search**" type
- The control should be followed by a **submit** button.
- The form must also include one hidden field to specify the domain for search and another one to specify that only that domain should be searched

```
<form method="get" action="http://www.google.com/search">
    <input type="search" name="q" size="30" maxlength="255">
    <input type="hidden" name="domains"
                value="https://developer.mozilla.org/en-US/">
    <input type="hidden" name="sitesearch"
                value="https://developer.mozilla.org/en-US/">
    <input type="submit" name="search" value="Search">
</form>
```


Mohamed Mukthar Ahmed

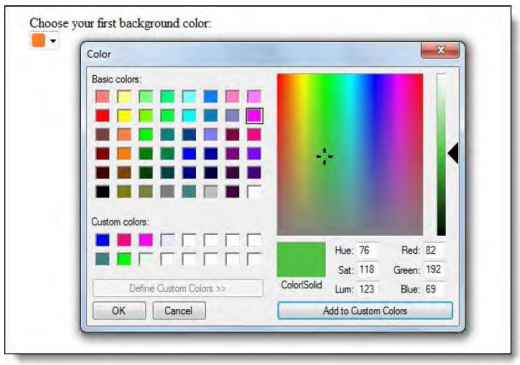


# Using Color Control

- The **input** element with the "**color**" type lets the user select a color from the color palette
- When the user selects the color the hexadecimal value for the color is returned

**The HTML for a color control**

**The Windows color palette**




```
background color:</label>
'firstcolor">
```

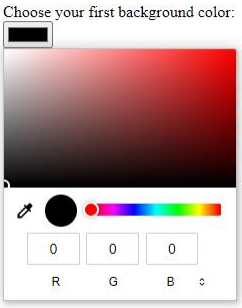
Mohamed Mukthar Ahmed

# Using Color Control

- The **input** element with the "**color**" type lets the user select a color from the color palette
- When the user selects the color the hexadecimal value for the color is returned

**The HTML for a color control**

```
<label for="firstcolor">Choose your first background color:</label>
<input type="color" name="firstcolor" id="firstcolor">
```




Mohamed Mukthar Ahmed



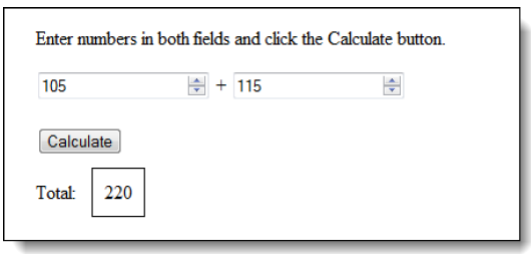
# Output Element

- The **output** element is used to display output data
- To show which fields the output element is associated with we use the **for** attribute

| Attribute | Description |
|---|---|
| for | Can be used to associate an output element with one or more form controls. |

**The HTML for a form that uses an output element**

```
<p>Enter numbers in both fields and click the Calculate button.</p>
<form onsubmit="return false">
    <input name="x" type="numb
    <input name="y" type="numb
    <input type="button" value
        onClick="result.value
    <br><br>
    Total: <output name="resul
</form>
```




Mohamed Mukthar Ahmed

# Progress & Meter Elements

- The **progress** and **meter** elements are also designed for output data
- Both these elements present data in **graphical form**
- By default the color used in these controls is **green/blue**

**Attributes for the progress and meter elements**

| Attribute | Description |
|---|---|
| `high` | The point at which the element's value is considered a high point. |
| `low` | The point at which the element's value is considered a low point. |
| `min` | The lower limit of the element. Typically, this will be 0 to represent 0%. |
| `max` | The upper limit of the element. |
| `optimum` | The point at which the element's value is considered optimum. |
| `value` | The current value of the element. |





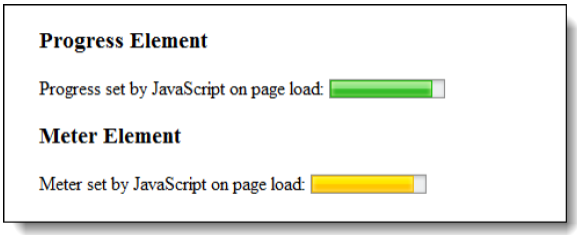
# Progress & Meter Elements

**The HTML for progress and meter elements**

```
<body onLoad="setProgressAndMeter()">
    <h3>Progress Element</h3>
    Progress set by JavaScript on page load:
    <progress id="progressBar" max="100" value="0"></progress>
    <h3>Meter Element</h3>
    Meter set by JavaScript on page load:
    <meter id="meterBar" max="100" value="0" optimum="50" high="60"></meter>
</body>
```

**The JavaScript that manipulates the progress and meter elements**

```
<script>
    function setProgressAndMeter() {
        var progress = document.getElementById("progressBar");
        setInterval(setPr
        progress.value +=
        var meter = docum
        setInterval(setPr
        meter.value += 10
    };
</script>
```



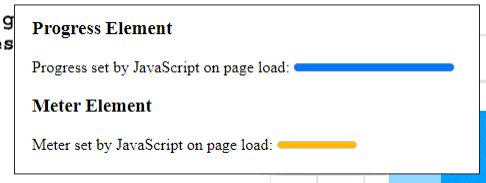
# Progress & Meter Elements

**The HTML for progress and meter elements**

```
<body onLoad="setProgressAndMeter()">
    <h3>Progress Element</h3>
    Progress set by JavaScript on page load:
    <progress id="progressBar" max="100" value="0"></progress>
    <h3>Meter Element</h3>
    Meter set by JavaScript on page load:
    <meter id="meterBar" max="100" value="0" optimum="50" high="60"></meter>
</body>
```

**The JavaScript that manipulates the progress and meter elements**

```
<script>
    function setProgressAndMeter() {
        var progress = document.getElementById("progressBar");
        setInterval(setProgressAndMeter, 500);
        progress.value += 10;
        var meter = document.g
        setInterval(setProgres
        meter.value += 10;
    };
</script>
```

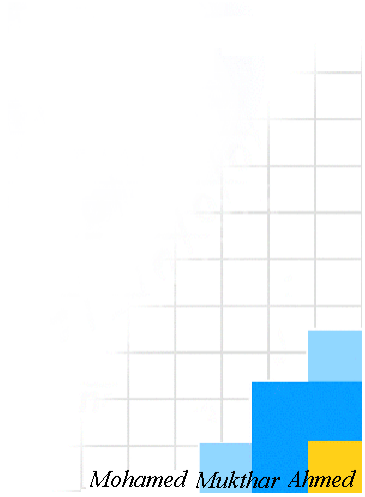



Mohamed Mukthar Ahmed

45

Mohamed Mukthar Ahmed

46




47