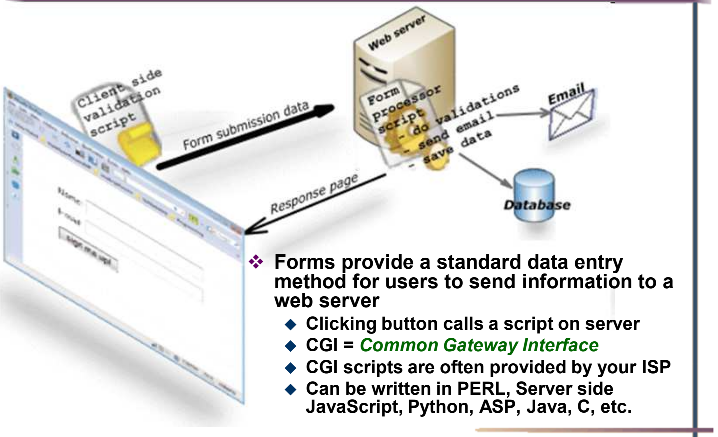
HTML Forms and Server Processing



- Forms provide a standard data entry method for users to send information to a web server
  - Clicking button calls a script on server
  - CGI = *Common Gateway Interface*
  - CGI scripts are often provided by your ISP
  - Can be written in PERL, Server side JavaScript, Python, ASP, Java, C, etc.

Dynamic HTML becomes HTML5

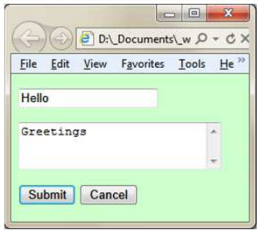
- D-HTML is the fusion of various web coding technologies that allows *dynamic* web content
  - D-HTML allows content presentation to change with user interaction or time
  - HTML alone will create static (unchangeable) pages
  - D-HTML = HTML + JavaScript + CSS + EM + DOM
- EM = Event Model
  - onclick = Click mouse on object event
  - onchange = Object state changes event
  - onkeydown = Key Down after entry event
- DOM = Document Object Model
  - document.frmCalc.txtEntry.value
  - <p id="result">
  - document.images.imgMain.src

DHTML - Mouse Events

EVENT	WORKS WITH	WHEN
onmouseover	Most elements	Mouse cursor over element
onmouseout	Most elements	Mouse cursor moves out of specifed element
onmousedown	Most elements	Mouse button down while cursor over element
onmouseup	Most elements	Mouse button released after clicking on element
onmousemove	Most elements	Mouse cursor moves while over the element
onclick	Most elements	Mouse button clicks on specified element
ondblclick	Most elements	Mouse button double clicks on specified element

Form Submission to Web Server

This form contains form elements Text Box and Text Area Code. When Submit button is pushed string sent to server using CGI for processing



Hello

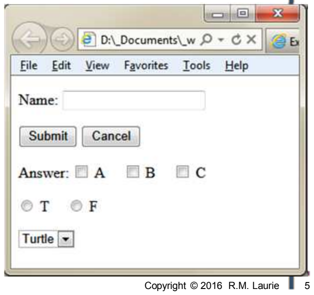
Greetings

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Text Demo</title>
  </head>
  <body style="background-color: #CCFFCC">
    <form name="frmBob" method="post" action="/cgi-bin/script.cgi">
      <p><input type="text" name="txtHi" value="Hello" size="20"
      maxlength="30"></p>
      <p><textarea rows="3" cols="25" name="txtareaGreet">
      Greetings</textarea></p>
      <p><input type="submit" name="btnOK" value="Submit">
      <input type="reset" name="btnCancel" value="Cancel"></p>
    </form>
  </body>
</html>
```

```
txtHi=Hello&txtareaGreet=Greetings&btnOK=Submit
```

HTML Forms and JavaScript Processing

- HTML Forms can be utilized to implement a (GUI) Graphical User Interface that interacts with JavaScript
  - Clicking a button or menu item triggers call to function
  - JavaScript functions can read input data from form elements
  - JavaScript functions can write output data to form elements
  - Formatting of form elements can be done using CSS styles
  - There are many form elements available in HTML5
    - Text box
    - Buttons: Submit, others
    - Check boxes
    - Radio buttons
    - Dropdown Menus



Name:

Submit Cancel

Answer: □ A □ B □ C

○ T ○ F

Turtle

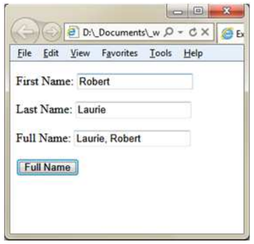
Form and Input Elements

- Form is a block level element
  `<form name="frmName" action="#"></form>`
  - name attribute is identifier of the form for older browsers
  - id attribute is identifier of the form for newer browsers & DOM
  - action specifies the Server script on web server to process the sent data; for JavaScript "#" works well
  - Don't forget to close your form elements
- Text input element is for single line text input
  `<input type="text" name="txtFirstName" tabindex="1">`
  - type="text" defines as a text box
  - name attribute is identifier of the form for older browsers
  - id attribute is identifier of the form for newer browsers & DOM
  - size attribute specifies character width of element
  - maxlength attribute specifies maximum number of characters entered
  - tabindex="1" is the first tab stop. Set to -1 to disallow tab
  - readonly="readonly" For results only not input
- Input button usually used to call function

```
<input type="button" name="btCalc" value="Calculate"
onclick="calculate()">
```

Old DOM Access utilizes document element name attribute for access of element

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Example using old DOM Specifications</title>
    <script type="text/javascript">
      function NameSwap()
      {
        var First = document.frmName.txtFirstName.value;
        var Last = document.frmName.txtLastName.value;
        document.frmName.txtFullName.value = Last + ", " + First;
      }
    </script>
  </head>
  <body>
    <form name="frmName" action="#">
      <p>
        First Name:
        <input type="text" name="txtFirstName" tabindex="1">
      </p>
      <p>
        Last Name:
        <input type="text" name="txtLastName" tabindex="2">
      </p>
      <p>
        Full Name:
        <input type="text" name="txtFullName" tabindex="-1" readonly="readonly">
      </p>
      <p>
        <input type="button" name="btnFullName" tabindex="3"
         value="Full Name" onclick="NameSwap();">
      </p>
    </form>
  </body>
</html>
```



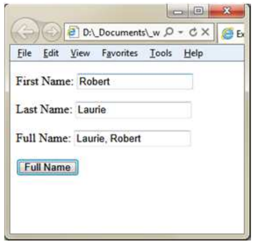
First Name: Robert

Last Name: Laurie

Full Name: Laurie, Robert

New DOM Access uses GetElementById to access form objects

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>
      GetElementById DOM Specifications
    </title>
    <script type="text/javascript">
      function NameSwap()
      {
        var First = document.getElementById("txtFirstName");
        var Last = document.getElementById("txtLastName");
        var Full = document.getElementById("txtFullName")
        Full.value = Last.value + ", " + First.value;
      }
    </script>
  </head>
  <body>
    <form id="frmName" action="#">
        <p>
          <label for="txtFirstName">First Name:</label>
          <input type="text" id="txtFirstName" tabindex="1">
        </p>
        <p>
          <label for="txtLastName">Last Name:</label>
          <input type="text" id="txtLastName" tabindex="2">
        </p>
       <p>
          <label for="txtFullName">Full Name:</label>
          <input type="text" id="txtFullName" tabindex="-1">
        </p>
        <p>
          <input type="button" id="btnSwap" tabindex="2"
         value="Full Name" onclick="NameSwap();">
        </p>
    </form>
  </body>
</html>
```



First Name: Robert

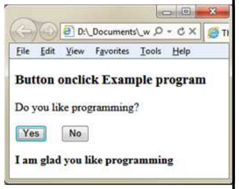
Last Name: Laurie

Full Name: Laurie, Robert

Note that element id attribute Is now the identifier. For old browser compatibility, sometimes name attributes included with id attributes

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>The Button Clicker</title>
    <script>
        function AnsYes()
        {
           document.getElementById('Answer').innerHTML
              = "<b>I am glad you like programming</b>";
        }
        function AnsNo()
        {
           document.getElementById('Answer').innerHTML
              = "<b>You will like it if you study</b>";
        }
    </script>
  </head>
  <body>
    <h3>Button onclick Example program</h3>
    <p>Do you like programming?</p>
     <p><button onclick="AnsYes()">Yes</button> &nbsp;
     &nbsp; <button onclick="AnsNo()">No</button></p>
     <p id="Answer">Click a button<p>
  </body>
</html>
```

Note in this example
a form is not utilized!
Clicking a button calls
a JavaScript function
to change the inner text
within an HTML element



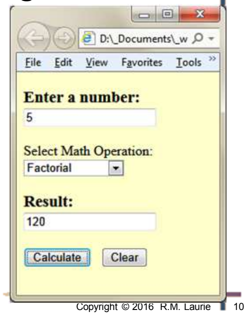
Button onclick Example program

Do you like programming?

I am glad you like programming

Select Menu

- ❖ Select menus use **select** and **option** elements
- ❖ Select menus work well with setting parameters
- ❖ Can be used to provide a Graphical User Interface (GUI) for JavaScript Programs
- ❖ This example utilizes a select menu to choose one of three functions:
  - ◆ Square
  - ◆ Square Root
  - ◆ Factorial
- ❖ Calculate button click calls Calculate() function
  - ◆ onclick is an event (Stay Tuned)



Enter a number:

5

Select Math Operation:

Factorial

Result:

120

Select Option Menu and Text Area

- ❖ Select Option Menu is drop down menu

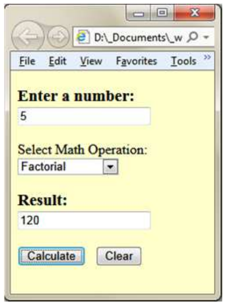
```
<select name="mnuMathOp" id="mnuMathOp">
    <option selected="selected">- Choose One -</option>
    <option>Square</option>
    <option>Square Root</option>
    <option>Factorial</option>
</select>
```

- ❖ <select> element attributes
  - ◆ name attribute is identifier of the form for older browsers
  - ◆ id attribute is identifier of the form for newer browsers & DOM
  - ◆ size attribute specifies options shown in menu
  - ◆ disabled can disable the menu
- ❖ <option> element attributes
  - ◆ selected is true or false if selected
- ❖ Text area element is for multi-line text input

```
<textarea rows="4" cols="30" id="txtAreaGreet"> Hello</textarea>
```

  - ◆ rows is the height
  - ◆ cols is the width
  - ◆ id is the identifier

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Select Option Example</title>
    <script src="Calculator.js"></script>
  </head>
  <body style="background-color: #FFFFCC">
    <form name="frmCalc" action="#">
      <h3>Enter a number:<br />
        <input type="text" id="txtEntry" size="20"></h3>
      <p>Select Math Operation:<br>
       <select id="mnuMathOp">
        <option selected="selected">
          - Choose One -</option>
        <option id="opSq">Square</option>
        <option id="opRt">Square Root</option>
        <option id="opFc">Factorial</option>
      </select> </p>
      <h3>Result:<br />
      <input type="text" id="txtResult" size="20"
        readonly="readonly"></h3>
      <p><input type="button" name="btCalc" value="Calculate"
      onclick="Calculate()"> &nbsp;
      <input type="reset" name="btClear" value="Clear" /></p>
    </form>
  </body>
</html>
```



Enter a number:

5

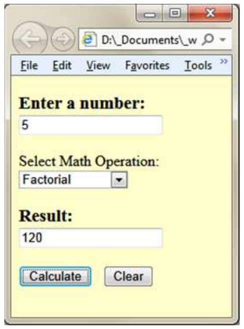
Select Math Operation:

Factorial

Result:

120

```
function Calculate()
{
  var Result, I, Selection;
  var Entry = document.getElementById("txtEntry");
  var Output = document.getElementById("txtResult");
  var OptSqur = document.getElementById("opSq");
  var OptRoot = document.getElementById("opRt");
  var OptFact = document.getElementById("opFc");
  Entry = parseFloat(Entry.value);
  if(OptSqur.selected)
    Result = Entry * Entry;
  else if(OptRoot.selected)
    Result = Math.sqrt(Entry);
  else if(OptFact.selected)
  {
    Result = 1;
    for(I = 1; I <= Entry; I++)
      Result = Result * I;
  }
  else
    window.alert("Select an Operation!");
  Output.value = Result;
  return;
}
```



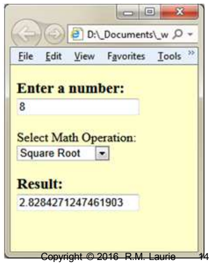
Enter a number:
5

Select Math Operation:
Factorial

Result:
120

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Select Option Example</title>
    <script src="Calculator.js"></script>
  </head>
  <body style="background-color: #FFFFCC">
    <form name="frmCalc" action="#">
      <h3>Enter a number:<br>
        <input type="text" id="txtEntry" size="20"
        onchange="Calculate()"></h3>
      <p>Select Math Operation:<br>
       <select id="mnuMathOp" onchange="Calculate()">
        <option selected="selected">
        - Choose One -</option>
        <option id="opSq">Square</option>
        <option id="opRt">Square Root</option>
        <option id="opFc">Factorial</option>
      </select> </p>
      <h3>Result:<br>
      <input type="text" id="txtResult" size="20"
        readonly="readonly"></h3>
    </form>
  </body>
</html>
```

Same JavaScript function is called but this code uses the onchange event for either select menu or entry text box



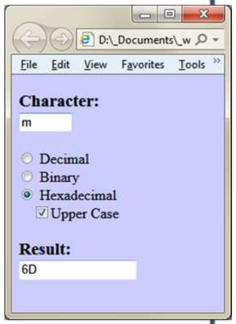
Enter a number:
8

Select Math Operation:
Square Root

Result:
2.8284271247461903

GUI Using Radio Buttons and Check Box

- Radio buttons and check boxes can enhance a GUI Form
- In this example you can type in a single ASCII character and convert it to the specified number system
- Note that the checkbox is enabled only when hexadecimal is selected
- The display changes when the character is changed, any radio button is clicked, or the check box is clicked (when enabled)



Character:
m

○ Decimal
○ Binary
● Hexadecimal
☑ Upper Case

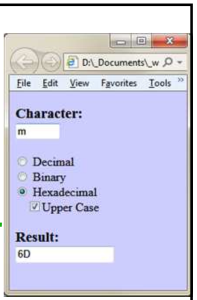
Result:
6D

Check Box and Radio Buttons

- Checkboxes use input element
  `<input type="checkbox" id="chkUpper" onclick="Convert()"`
  `disabled="disabled">`
  - type="checkbox" defines as a check box
  - name attribute is identifier of the form for older browsers
  - id attribute is identifier of the form for newer browsers & DOM
  - tabindex="1" is the first tab stop. Set to -1 to disallow tab
  - checked ="checked" initializes to checked
  - disabled="disabled" disallows changing
- Radio buttons use input element and has same name to interlink

  `<input type="radio" name="radConv" id="radDec"`
  `onclick="Convert()">`
  - type="radio" defines as a radio button
  - name attribute is required if link buttons to allow only one selection
  - id attribute must be unique for the page
  - tabindex="1" is the first tab stop. Set to -1 to disallow tab
  - checked ="checked" initializes to checked
  - disabled="disabled" disallows changing

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Convert Character</title>
    <script src="baseConvertor.js"></script>
  </head>
  <body style="background-color: #CCCCFF">
    <form id="frmConvert" name="frmConvert" action="#">
      <h3>Character:<br>
      <input type="text"  id="txtEntry" value="0" size="4"
        maxlength="1" onkeyup="Convert()"></h3>
      <p><input type="radio" name="radConv" id="radDec"
      onclick="Convert()"> Decimal<br>
      <input type="radio" name="radConv" id="radBin"
      onclick="Convert()"> Binary<br />
      <input type="radio" name="radConv" id="radHex"
      onclick="Convert()"> Hexadecimal<br>
       &nbsp; &nbsp; <input type="checkbox" id="chkUpper"
       onclick="Convert()" disabled="disabled">Upper Case</p>
      <h3>Result:<br>
      <input type="text" id="txtResult" size="16"
      maxlength="10"></h3>
    </form>
  </body>
</html>
```



Character:

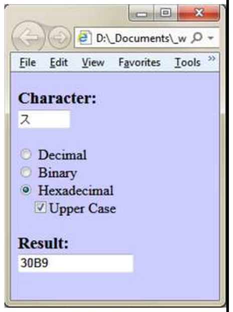
m

○ Decimal
○ Binary
● Hexadecimal
☑ Upper Case

Result:

6D

```
function Convert()
{
  var Result="", KeyCode;
  var Entry = document.getElementById("txtEntry");
  var Dec = document.getElementById("radDec");
  var Bin = document.getElementById("radBin");
  var Hex = document.getElementById("radHex");
  var Upper = document.getElementById("chkUpper");
  var Output = document.getElementById("txtResult");
  KeyCode = Entry.value.charCodeAt(0); // Unicode
  Upper.disabled=true;
  if(Dec.checked)
    Result = KeyCode.toString(10);
  else if(Bin.checked)
    Result = KeyCode.toString(2);
  else if(Hex.checked)
  {
    Upper.disabled=false;
    Result = KeyCode.toString(16);
    if(Upper.checked)
      Result = Result.toUpperCase();
    else
      Result = Result.toLowerCase();
  }
  Output.value = Result;
  return;
}
```



Character:

ス

○ Decimal
○ Binary
● Hexadecimal
☑ Upper Case

Result:

30B9