Lecture 6

JAVASCRIPT

Marie Brennan

Learning Outcomes

- You will be able to:
  - *Place JavaScript in appropriate places*
    - In event handler attributes (rollover, form onSubmit())
    - In <script> tags
    - Place reusable functions in separate text files ("*.js")
      - *and load them from HTML*
  - *Write form validation functions*
    - To prevent invalid data being sent to server

What is the relationship between “Java” and “JavaScript”?

- *None at all !*

What is JavaScript?

- Web page scripting language
- JavaScript little in common with Java
- A JavaScript program
  - *either embedded in a Web page*
  - *or loaded by a web page*
- a HTML tag introduces it

What kind of things is JavaScript used for?

- *Validating forms, e.g.*
  - error message if enter text in a telephone number field
- *Interactive web pages*
  - Pop-up windows / mouse-over button highlights
- *Calculations based on selections on a form*
- *Often used as part of front-end for eCommerce*

LOCATION OF JAVASCRIPT

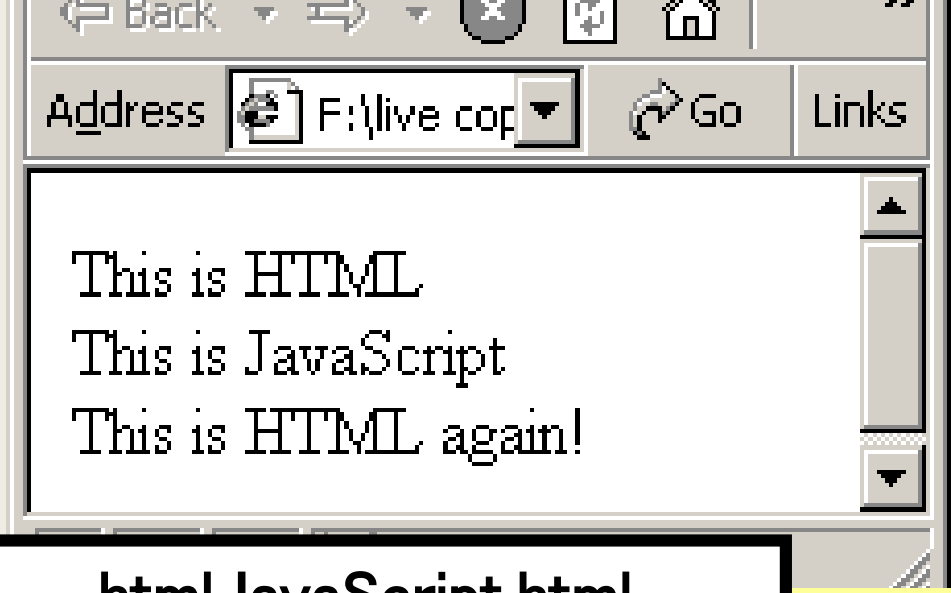
Where is JavaScript code located?

- Located in three possible places
  - *(1) Inside **<script> </script>** tags*
    - inside an HTML page
  - *(2) as an **event handler** (e.g. link / button mouseover)*
    - inside an HTML page
  - *(3) Or in a **separate text file** ("<actions>.js")*
    - A <script> tag in HTML page with file "src" attribute
- Where is the JavaScript executed?
  - *By the web browser*
  - *So it is a "client-side" scripting language*

Location 1
Inside <script> elements in HTML file

- <script> elements can be placed in the HEAD or BODY sections on an HTML document
- JavaScript statements can be placed inside the open-close SCRIPT tags

Dynamic HTML...



htmlJavaScript.html

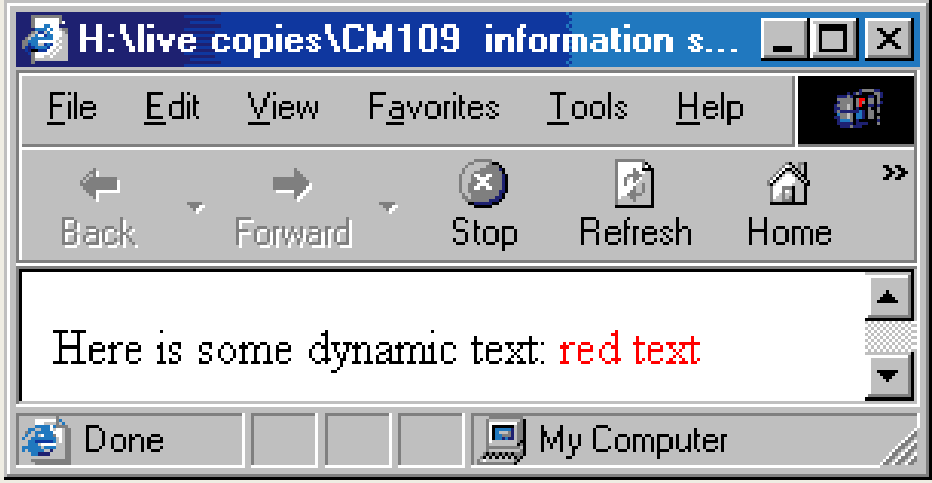
```
<html>
<body>
<p>
This is HTML
<br>
  <script type="text/javascript">
        document.write( "This is JavaScript" );
  </script>
<br>
This is HTML again!
</p>
</body>
</html>
```

- Previous example
  - *Seems no big deal*
  - *could have done this without JavaScript.*
- However it demonstrates **<script>** tag
  - *Everything inside is interpreted as JavaScript*
  - ***Illustration of JavaScript statement***

    document.write()
    - Outputs String-text into to the HTML page at runtime

JavaScript inside SCRIPT tags

dynamicText.html




```
<html>
<body>
<p>
Here is some dynamic text:

<script type="text/javascript">
      var message = "<font color='red'>red text</font>" ;
      document.write(message  );
</script>

</p>
</body> </html>
```

A popup window


```
<html>
<body>
<h1>
   Hello world program in JavaScript
</h1>
<script type="text/javascript">
   // display popup window saying "Hello World!"
      alert( "Hello World !" );
</script>
</body>
</html>
```

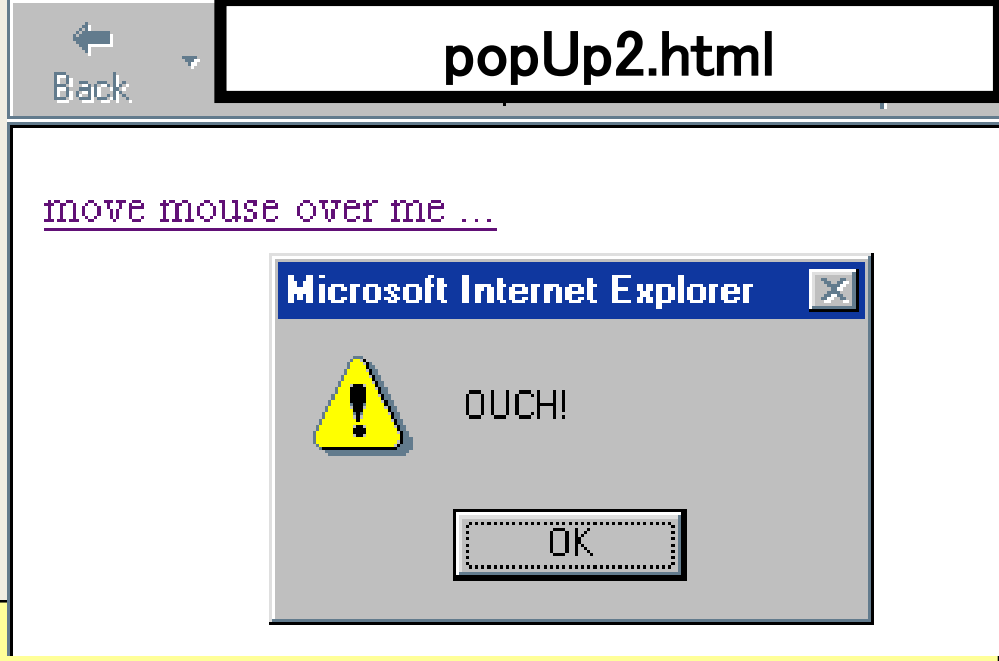
Popup window – alert(“ “)

- An “alert” is a popup window displaying a message
- All the user can do is click OK to acknowledge the message
  - *No user input apart from this is returned to JavaScript*
- Useful for
  - *Error messages*
  - *Debugging – can alert yourself to value in a variable*
  - *Simple messages, such as when events occur*
    - E.g. Your order has been successfully processed and goods are being dispatched

- In the previous examples
  - *<script> and </script> tag pair are placed in the HTML file BODY*
  - *Inside the tags, 2 JavaScript statements are written*
    - A String variable is defined
    - The String is written to the HTML document

- Not recommended in BODY !!!

Location 2
As an event handler




```
<html>
<script type = "text/javascript">
</script>
</head>
<body>
      <a     href="#"
             onMouseOver="alert('OUCH!');" >
      move mouse over me ...
      </a>
</body>
</html>
```

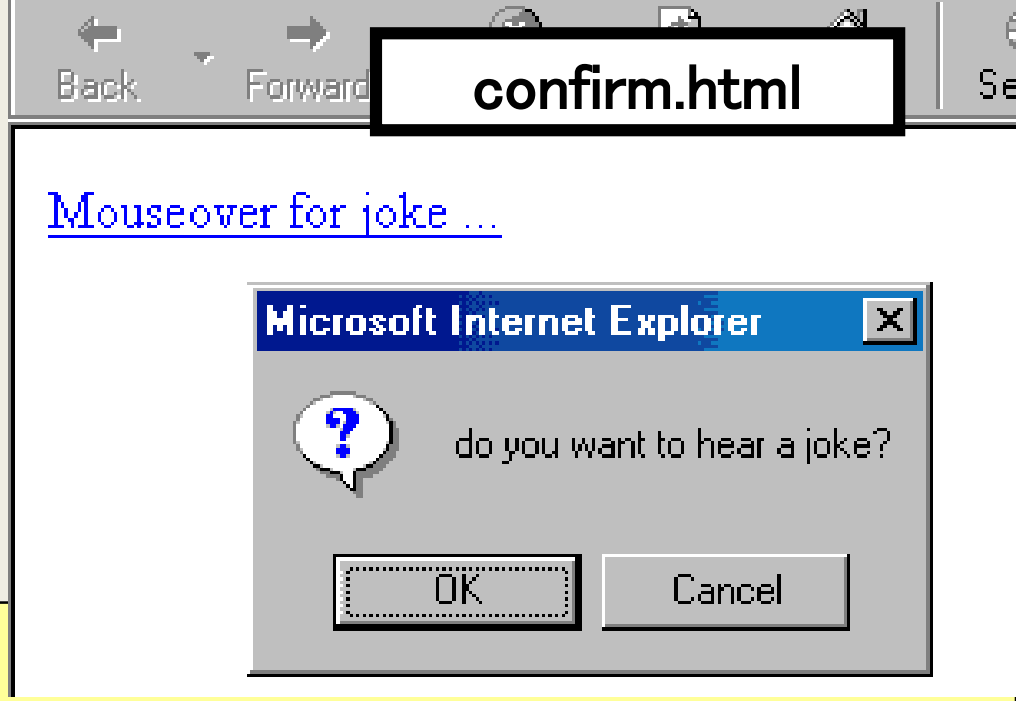
Event handler

- The anchor text **`"move mouse over me ..."`**
  - *a dummy anchor “#”*
  - *Used since anchors are DOM objects that can have* ***event handlers*** *such as for mouse over*

- The statement to be executed when a mouse cursor moves over the anchor text is:
  - ***`onMouseOver = "alert('OUCH!');"`***
  - *JavaScript to display a pop-up window with the text “OUCH!”*

DOM = document object model

Popup OK/Cancel window

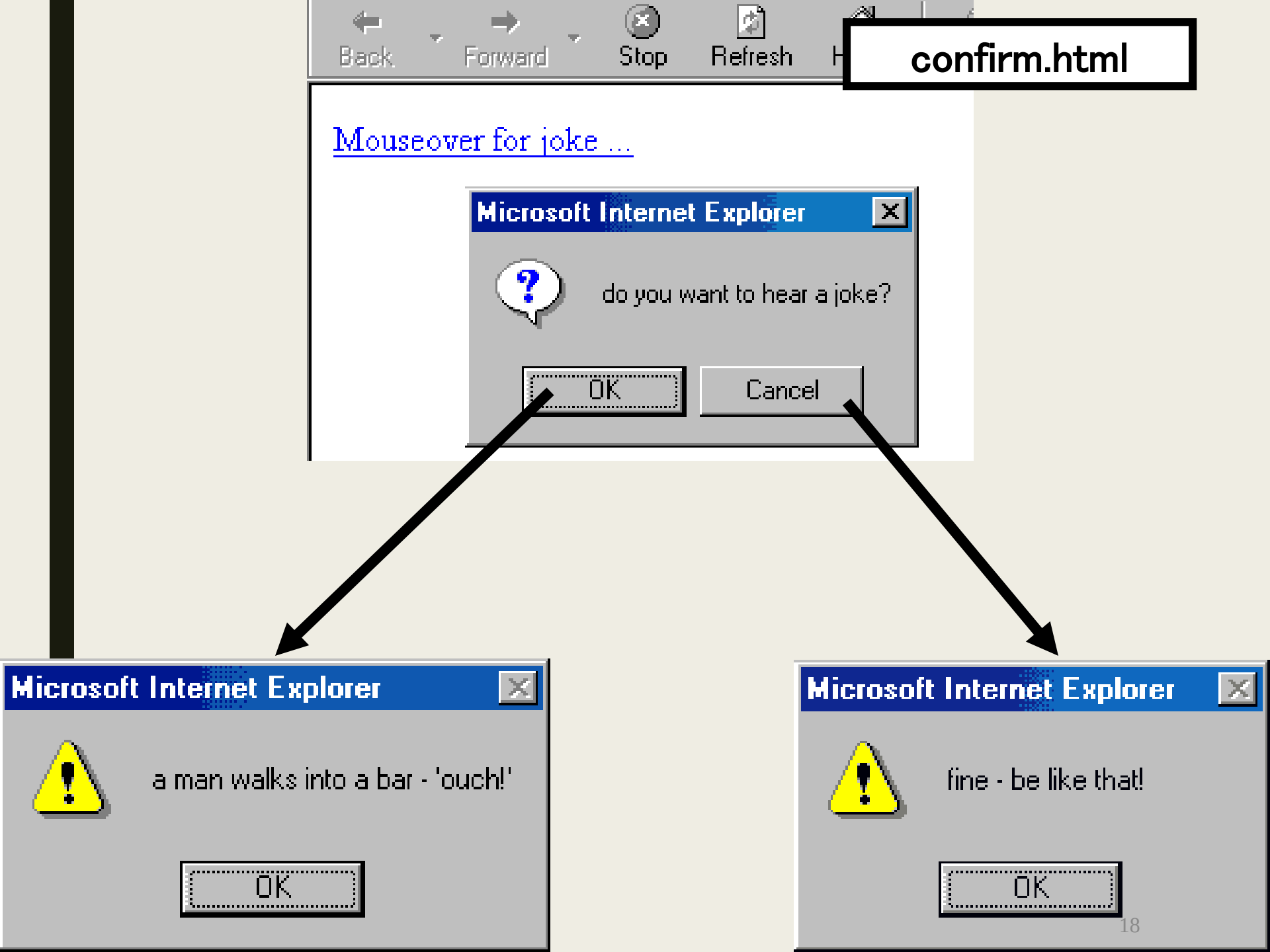



```
<html>
<script type="text/javascript"> >
    ************ JavaScript statements here ************
</script>
</head>

<body>
    <a href="#" onMouseOver="popupJoke();" >
                Mouseover for joke ... </a>
</body>
</html>
```

Event handler

Function


Back
Forward
Stop
Refresh
confirm.html
Mouseover for joke ...
Microsoft Internet Explorer
do you want to hear a joke?
OK
Cancel
Microsoft Internet Explorer
a man walks into a bar - 'ouch!'
OK
Microsoft Internet Explorer
fine - be like that!
OK

********** JavaScript statements here **********

confirm.html

```
function popupJoke()
{
      // popup window and store reply
      var okayChosen = confirm("do you want to hear a joke?");

      // respond to reply
      if( okayChosen = = true )
      {
            alert("a man walks into a bar - 'ouch!' ");
      }
      else
      {
            alert("fine - be like that!");
      }
}
```

Confirm () window popup
- OK/Cancel

- "Alert" windows only give user "OK" button to respond to
- "Confirm" windows offer "OK" and "Cancel"

  Function returns Boolean **true** is OK was chosen and **false** if Cancel was chosen

  We can store return value in a variable

  - And use IF statements to respond appropriate to user interaction

Location 3

In an external text file

externalFile.html


```
<html>
<body>
<h1>Heading 1</h1>

<script      src = "js/new_heading.js"
                   type = "text/javascript">
</script>

</body>
</html>
```

- Contents of external file
  - *"new_heading.js"*
  - *Javascript files have .js extension*
  - *Stored in external folder called js*
- JavaScript statements
  - *First line is a comment*
    - JavaScript comments are like Java "//" and "/* ..... */"
  - *Second line creates a String variable* ***newHeading***
  - *Third line writes String into body of document*

```
// JavaScript to write HTML into browser document
var newHeading = "<h1>This line from external file</h1>";
document.write( newHeading );
document.writeln(newHeading); //goes to new line
```

new_heading.js

Questions

- What does an alert() do?
- What does confirm() do?
- JavaScript can be used in 3 different places. Name them.
- What does document.write() do? What is it similar to in Java?
- How is a variable declared in JavaScript?
- How is a function declared in JavaScript?
- What is the file extension of a JavaScript file?

JavaScript language reference

JavaScript Grammar

- Variables
- Operators
- Expressions

  total=100;
- Statements

  if (total>100) {statements;} else {statements;}
- Functions / Methods

Variables

- JavaScript is Case Sensitive

  myvariable not the same as Myvariable

- Variable names must start with a letter or underscore ("_"); subsequent characters can also be digits (0-9).
- Create a global variable by assigning it a value:

  myVariable = 5;

- Create a local variable using the keyword var:

  var myLocalVariable = 5;

Data Type

- JavaScript is a loosely typed language.
- Data types are converted automatically as needed during script execution

- String: - delineated by single or double quotation marks “Hello World”
- Boolean:- can be True or False
- Object: - myObj = new Object();

Operators

- Arithmetic:

  `+  -  *  /  %  ++  --`

- Comparison:

  `==  !=  >  >=  <  <=`

- Boolean:

  `&&  ||  !`

- String
- Numbers:

  Integers can be decimal, octal or hex

- Assignment:

  `=  +=  -=  *=  /=  &=  |=`

Conditional Statements

```
if (condition)
  {
  statements1;
  }
else
  {
  statements2;
  }
```

```
switch (expression){
    case label :
        statement;
        break;
    case label :
        statement;
        break;
    ...
    default : statement;
}
```

Loops

- FOR

```
for (initial-statement; test; increment)
 { statements; }
```

```
for (i=0; i<10; i++)
 { statements; }
```

- WHILE

```
do
 { statements;}
while (condition)
```

```
while (condition)
 { statements; }
```

- “break” and “continue”

Functions
- JavaScript “methods”

- Functions / methods

  Used to group together JavaScript statements

  - Keep things tidy by defining functions in separate file
  - Can re-use functions easily
  - Can make function an event-handler

Example of function

- Function “myFunction()” defined in <HEAD>

```
<html>
<head>
  <script type = "text/javascript">
      function myFunction()
      {
          document.write("Welcome to my page!<br>");
          document.write(">>> This isJavaScript!<br>");
      }
  </script>

</head>
```

Note – use of HTML tags in Javascript

Example of a function

- Function called from <BODY>

Functions !

Welcome to my page!
>>> This is JavaScript!
Welcome to my page!
>>> This is JavaScript!
Welcome to my page!
>>> This is JavaScript!

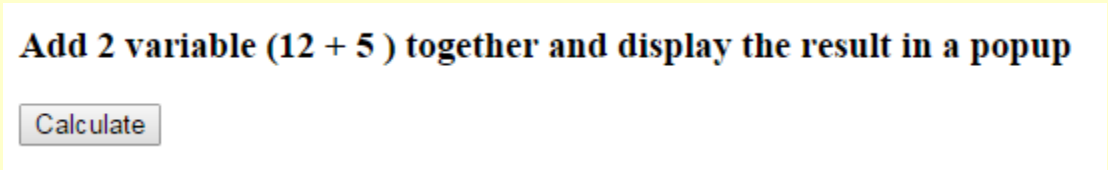
function.html

```
<body>
<h1> Functions ! </h1>
      <script type = "text/javascript" >
            myFunction();
            myFunction();
            myFunction();
      </script>
</body>
</html>
```

Function with Event Handler

```
<script type = "text/javascript">
function calc(){
        var x = 12;
        var y = 5;

        alert(x + y);……
```

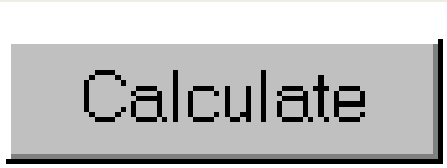



```
----------------------------------------
<h3>Add 2 variable (12 + 5 ) together and display the result in a
popup </h3>

        <button onclick="calc()">Calculate</button>
```

calculate.html

Function with Event Handler



- Click on button
- To run method **calc()**

```
function calc()
{
        var x = 12;
        var y = 5;
        var result = x + y;
        alert( result );
}
```

calculate.html

- Which pops-up answer alert

Events and event handlers ...

EventHandler	Called when . . .
onClick	User clicks on form element or link
onChange	User changes value of text, textarea, or select element
onFocus	User gives form element input focus
onBlur	User removes input focus from form element
onMouseOver	User moves mouse pointer over a link or anchor
onMouseOut	User moves mouse pointer off of link or anchor
onSelect	User selects form element's input field
onSubmit	User submits a form
onReset	User resets a form
onResize	User resizes the browser window
onLoad	User loads the page in the Navigator
onUnload	User exits the page

Event Handlers within HTML

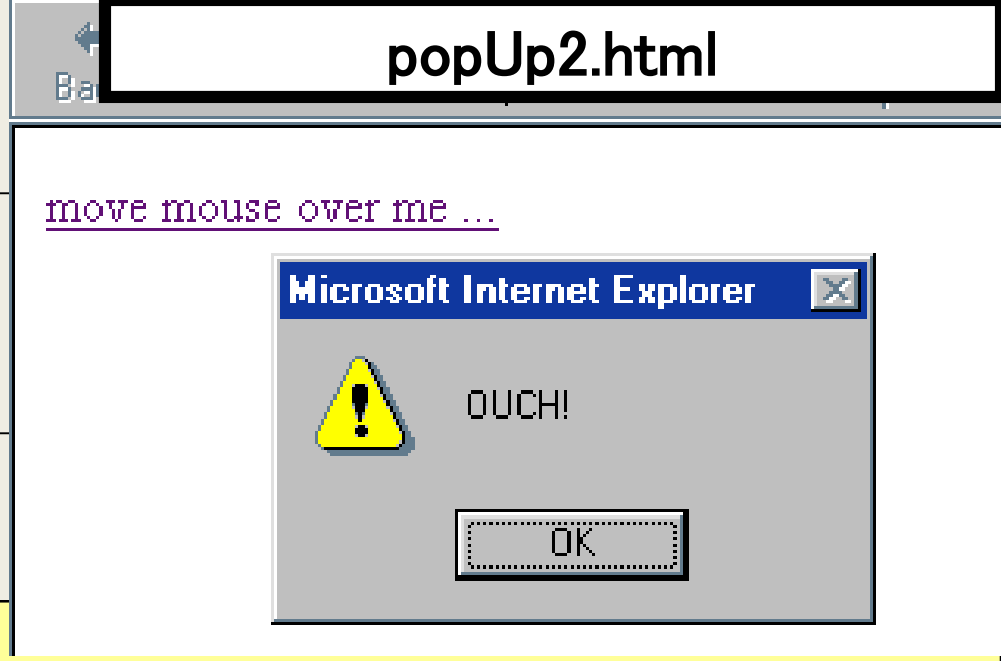
- The event handler specifies which JavaScript code to execute.
- Often, event handlers are placed within the HTML tag which declares the object on which the event acts.
- Example

```
<a href="url"
onMouseOver="popupFunc();">
```

- DEMO

onMouseOver

popUp2.html




```
<html>
</head>

<body>
      <a     href="#"
          onMouseOver="alert('OUCH!');" >
      move mouse over me ...
      </a>
</body>
</html>
```

Event handler

onMouseOver

```
<!DOCTYPE html>

<html>


<head>

<title>Javascript added to head of HTML file</title>

<script>
    @import "css/style.css";
</script>

<script type= "text/javascript">

/* WRITE YOUR FUNCTION IN HERE */

var eyesLeft = new Image();
eyesLeft.src = "images/eyesleft.gif";


var eyesRight = new Image();
eyesRight.src = "images/eyesright.gif";

</script>
</head>
<body>
<h1>Move your mouse over the image</h1>

<a href = "#" onMouseOver = "image.src = eyesRight.src"
              onMouseOut = "image.src = eyesLeft.src">

              <img src = "images/eyesleft.gif" name = "image" id = "image" width = "75" height = "75"> </a>


</body>
</html>
```

onClick

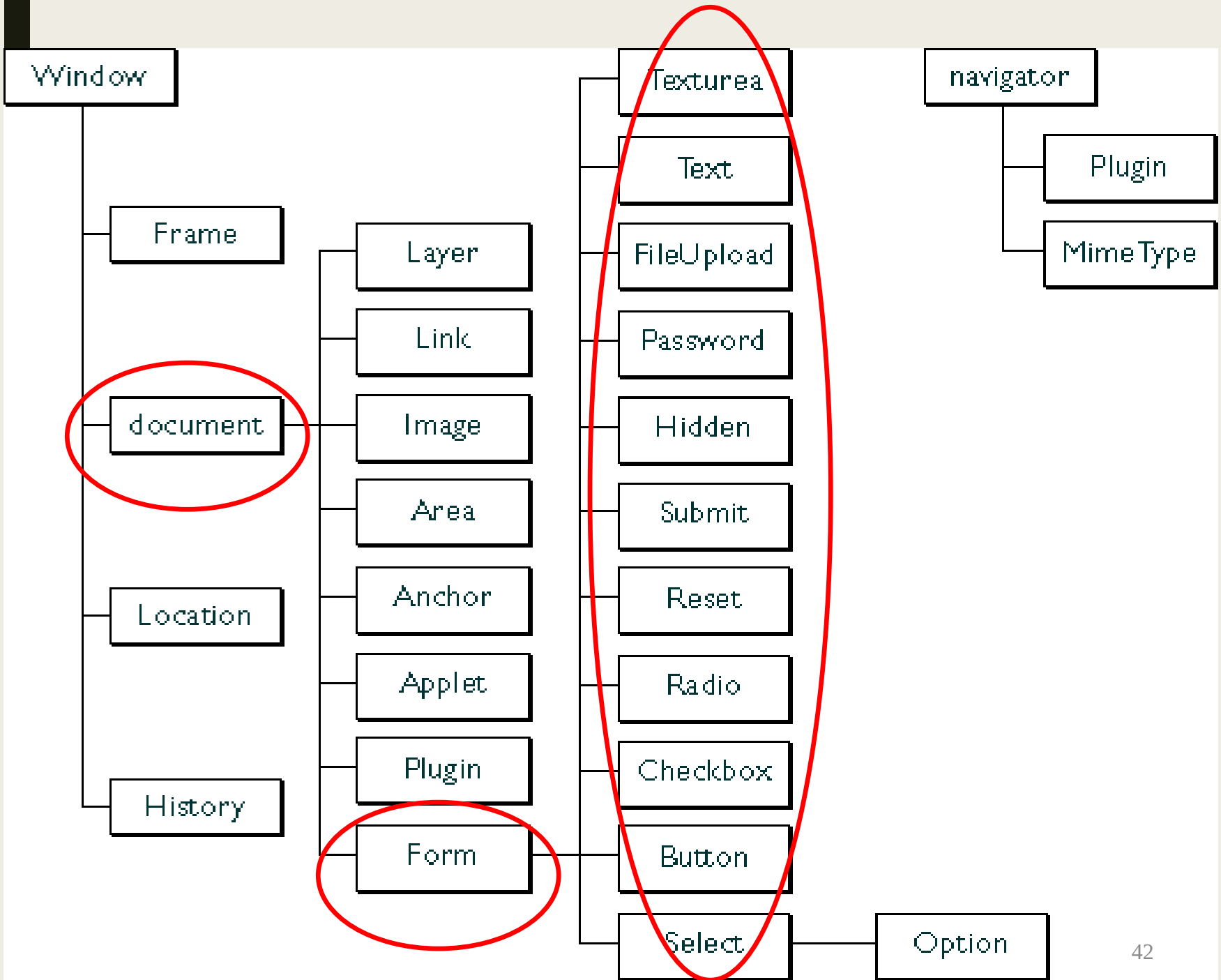
calculate.html


```
function calc()
{
        var x = 12;
        var y = 5;
        var result = x + y;
        alert( result );
}
```

```
<h1> Calculation in Javascript </h1>
<h3>Add 2 variable (12 + 5 ) together and display the result in a popup </h3>
<button onClick="calc()">Calculate</button>
```

Document Object Model (DOM)

- DOM is a hierarchical representation of all the components of a web page
- Top-level object is window
- Document object is child of window
- Document object has forms, links, images etc. array of components
- Form has its own array of form components


Window
Frame
document
Location
History
Layer
Link
Image
Area
Anchor
Applet
Plugin
Form
Texturea
Text
FileUpload
Password
Hidden
Submit
Reset
Radio
Checkbox
Button
Select
Option
navigator
Plugin
MimeType

Document Object Model (DOM)

- OBJECTs have **properties** and **methods**
- **Properties** of document in web browser can be altered directly from JavaScript

  E.g.

```
document.bgcolor = "red";
```

  Using the **dot notation** to represent properties and methods

  E.g.

  window.onload()

  document.write("The message is " + message);

  document.writeln("Hello " + name);

some DOM methods

- Window Object
  - close() closes the current browser window
  - **alert(message)**
  - **confirm(question)** ask a yes/no question
  - resizeTo (width,height)
- Document Object
  - write(value) .....document.write(“ ABC”);
  - writeln(value)..... document.writeln(“abc”);
  - getElementById( ) ... document.getElementById(“..”);

Using

- getElementById()
- innerHTML

DOM - . innerHTML

- Ever wondered how you could change the contents of an HTML element?
- Maybe you'd like to replace the text in a paragraph to reflect what a visitor has just selected from a drop down menu
- By manipulating an element's **innerHTML,** you'll be able to change your text and HTML as much as you like.

DOM .getElementById( )

- Each HTML element has an *innerHTML* property - defines both the HTML code and text which is between that element's opening and closing tag.
- By changing an element's *innerHTML* after some user interaction, can make pages more interactive .
- Must give the element you wish to change an **id**
- With that **id** in place, you will be able to use the **getElementById( )** function.
- When you have that set up, you can now manipulate the text of an element

•Change text inside **<p> tag**

```
<script type="text/javascript">
function changeText(){
var  newtxt = document.getElementById("welcome");
newtxt.innerHTML = "Goodbye Joe";
}
</script>

  ..........................
<p id = "welcome">Welcome to the site  Joe </p>
      <input type="button"  onclick="changeText();"
      value="Change Text">
```

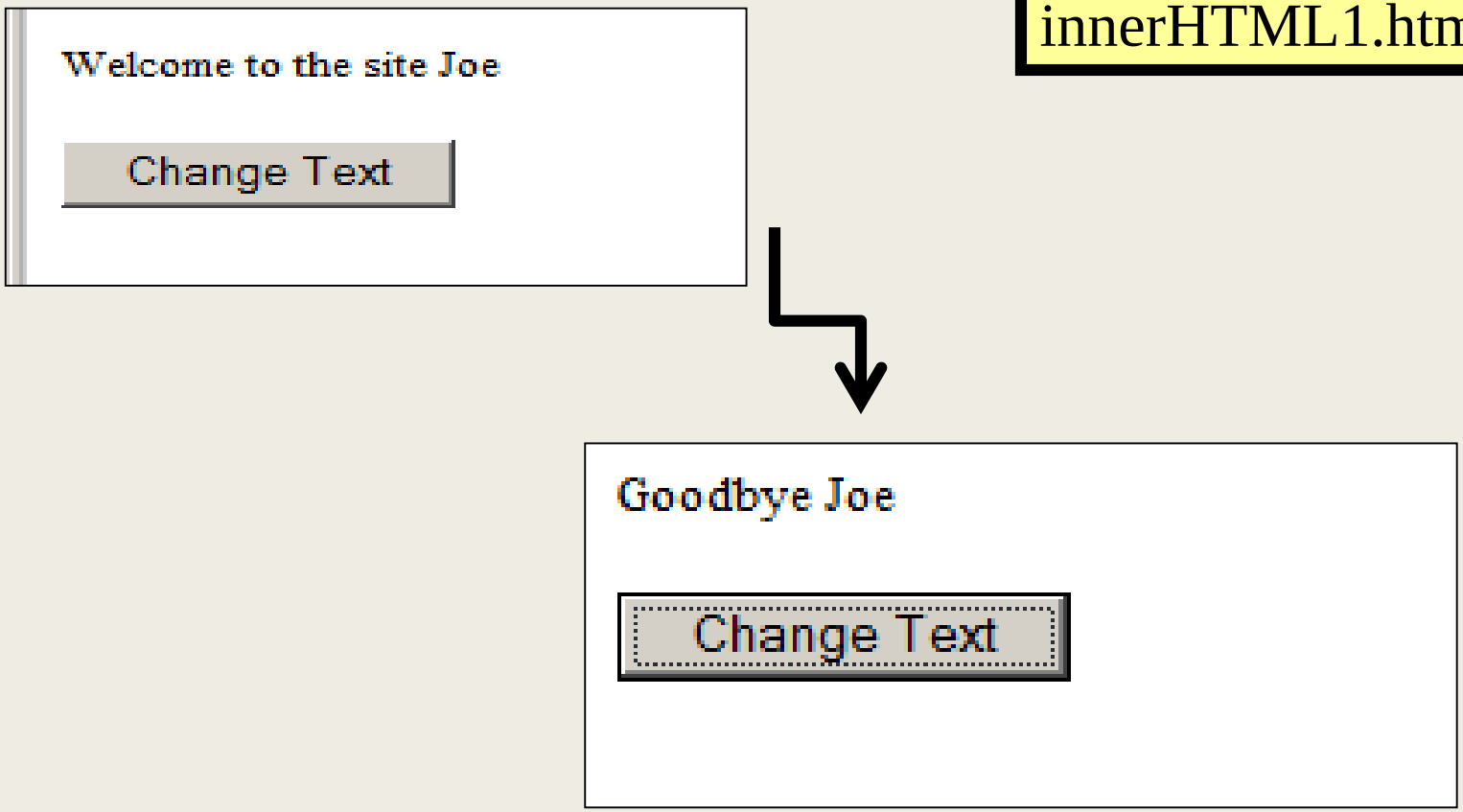
- JavaScript locates the html tag that has a unique ID associated with it.
  - *document.getElementById(“welcome”);*
- Must write - document.getElementById( ) – exactly, as shown
- Note the dot .
- “welcome” is the name of the ID you are looking for – it is sent to the method (getElementById(“welcome”)) as an argument

- Assign a text string to this property

```
var  newtxt = document.getElementById("welcome");
newtxt.innerHTML = "Goodbye Joe";
```

- Can include tags and text in innerHTML
- " " placed around tags, + (concatenator) used to join parts of strings

innerHTML1.html



Update HTML based on user input

```
<script type="text/javascript">
function changeText2()
{
var userInput  =   document.getElementById(“userInput”).value;
document.getElementById(“welcome”).innerHTML = userInput;
}
</script>  .....
<p id = “welcome” >Welcome to the site  Joe </p>
<input type='text'  id=“userInput”  value='Enter Text Here' >
        <input type='button'  onClick=“changeText2()”
          value='Change Text'>
```

//Take the value the user has input and put it into a variable called userInput

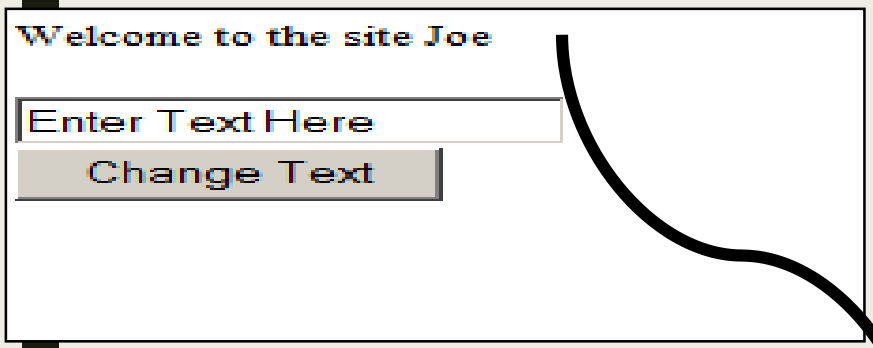
```
var userInput  =   document.getElementById("userInput").value;
```

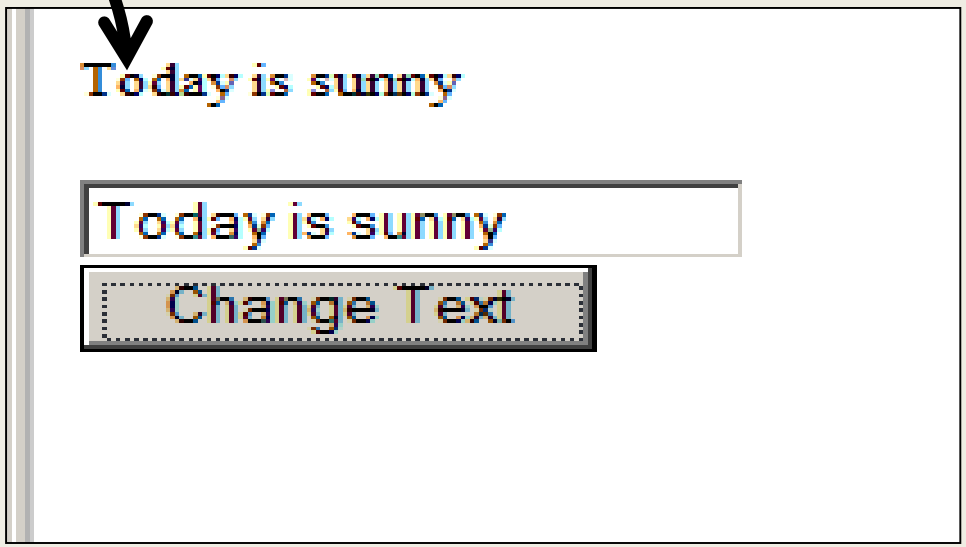
// Assign the contents of "userinput" to "welcome" by changing its innerHTML

```
document.getElementById("welcome").innerHTML =
userInput;
```

innerHTMLuserInput.html

innerHTMLuserInput.html





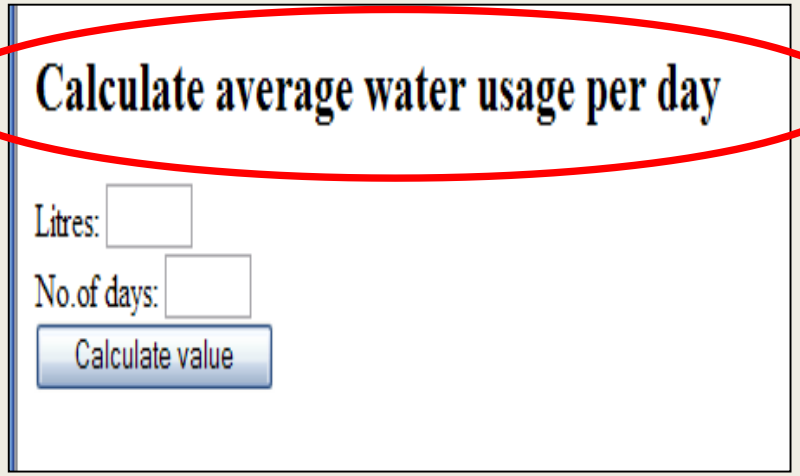
Show today's date

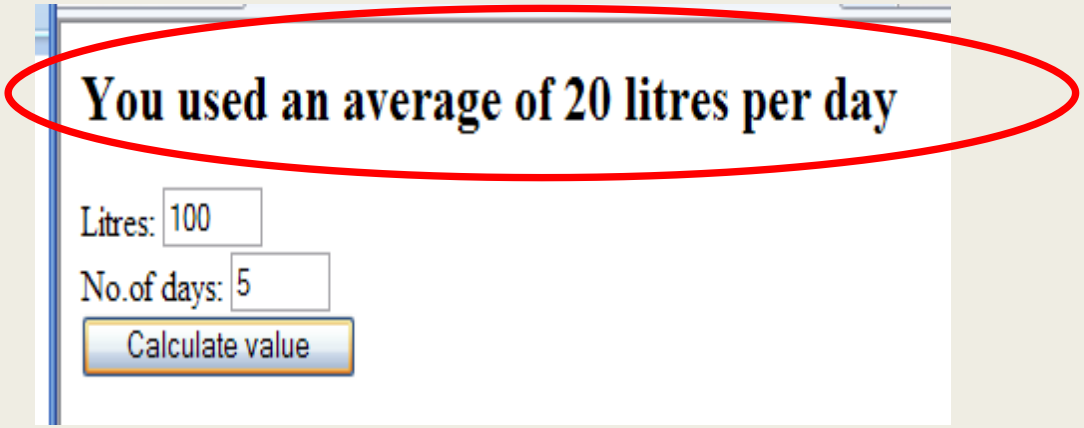
```
<!DOCTYPE html>
<html>
<script>
function callDate(){
        var d = new Date();
        document.getElementById("demo").innerHTML = d.toDateString();
}
</script>
<body onLoad="callDate();">
<h2> Display the current date </h2>
<p id="demo"></p>
</body>
</html>
```

date.html

Using JavaScript for Calculations

Water Usage Calculation





57

Calculation Question 1

- Water usage Example
- Create a webpage with 2 input boxes and a button.
- When the user presses the button, a function called **usage()** is called**,** which will compute the average number of litres of water used per day.
- The average is calculated by dividing the litres by the days **(litres / days**).
- Display the result in the form.

58

HTML - using ids

WaterUsage.html

```
<body>
<h2><div id="output">Calculate average water usage per day</h2></div>

Litres: <input type="text" size= "4" id="litres"><br/>
No.of days: <input type="text" size= "4" id="days"><br/>

<input type="button"  value="Calculate value" onclick="mpg()" />
```


59

JavaScript

WaterUsage.html

```
<script type="text/javascript">
function mpg()
{
var days = document.getElementById("days").value;
var ltrs = document.getElementById("litres").value;
var avg = ltrs/days;

document.getElementById("output").innerHTML = "You used an average of " + avg + " litres per day";

}
</script>
</head>
```


60

Recap on innerHTML property

- Can be used to modify your document's HTML on the fly.
- When you use innerHTML, you can change the page's content without refreshing the page. This can make your website feel quicker and more responsive to user input.
- The innerHTML property is used along with getElementById within your JavaScript code to refer to an HTML element and change its contents.

JavaScript Tutorials

- www.w3schools.com
- www.quackit.com/javascript