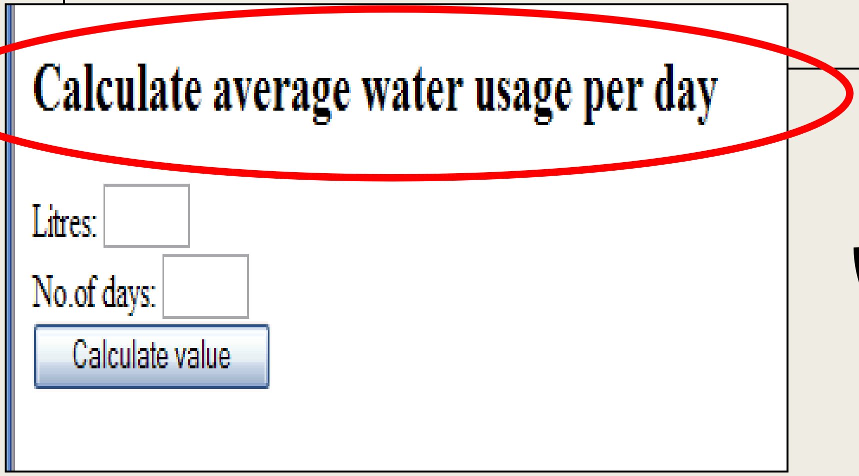
LECTURE 8
JAVASCRIPT

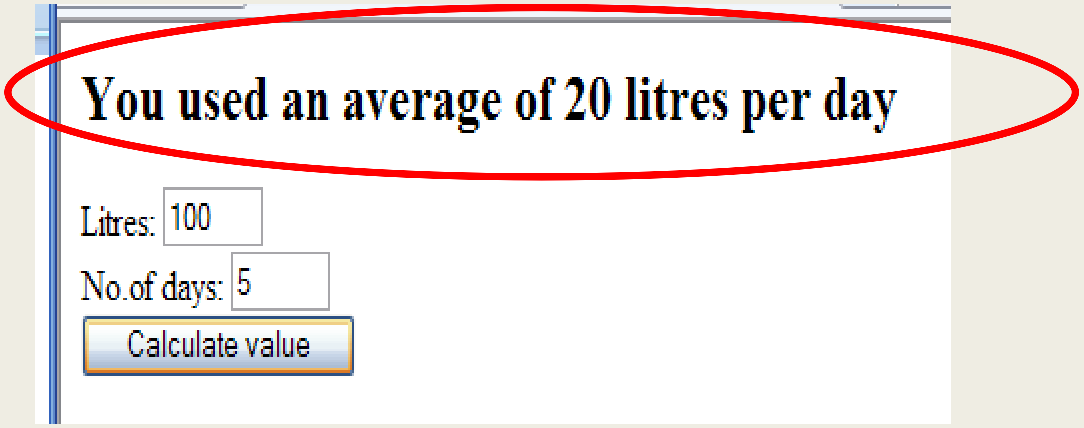
event Handlers/arrays/forms

Water Usage Calculation


Calculate average water usage per day

Litres:

No.of days:


You used an average of 20 litres per day

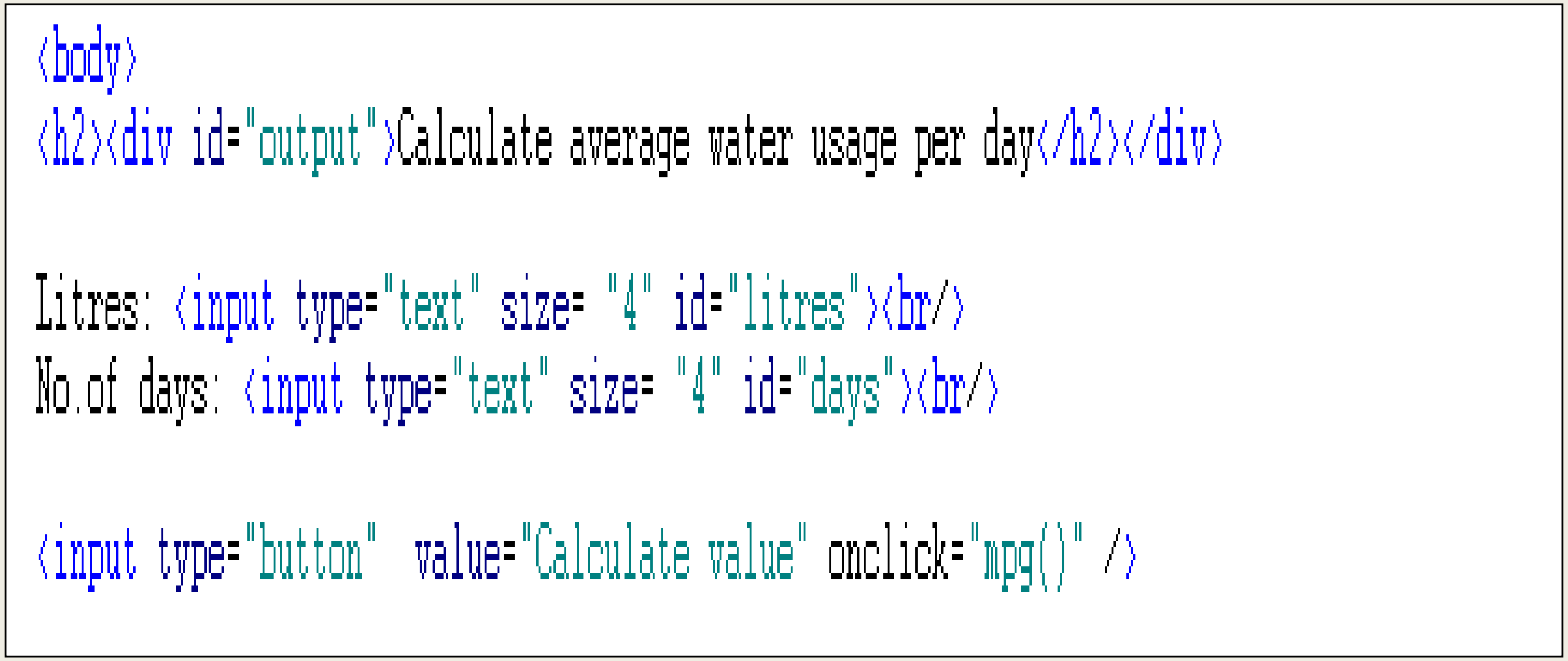
Litres: 100

No.of days: 5

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.

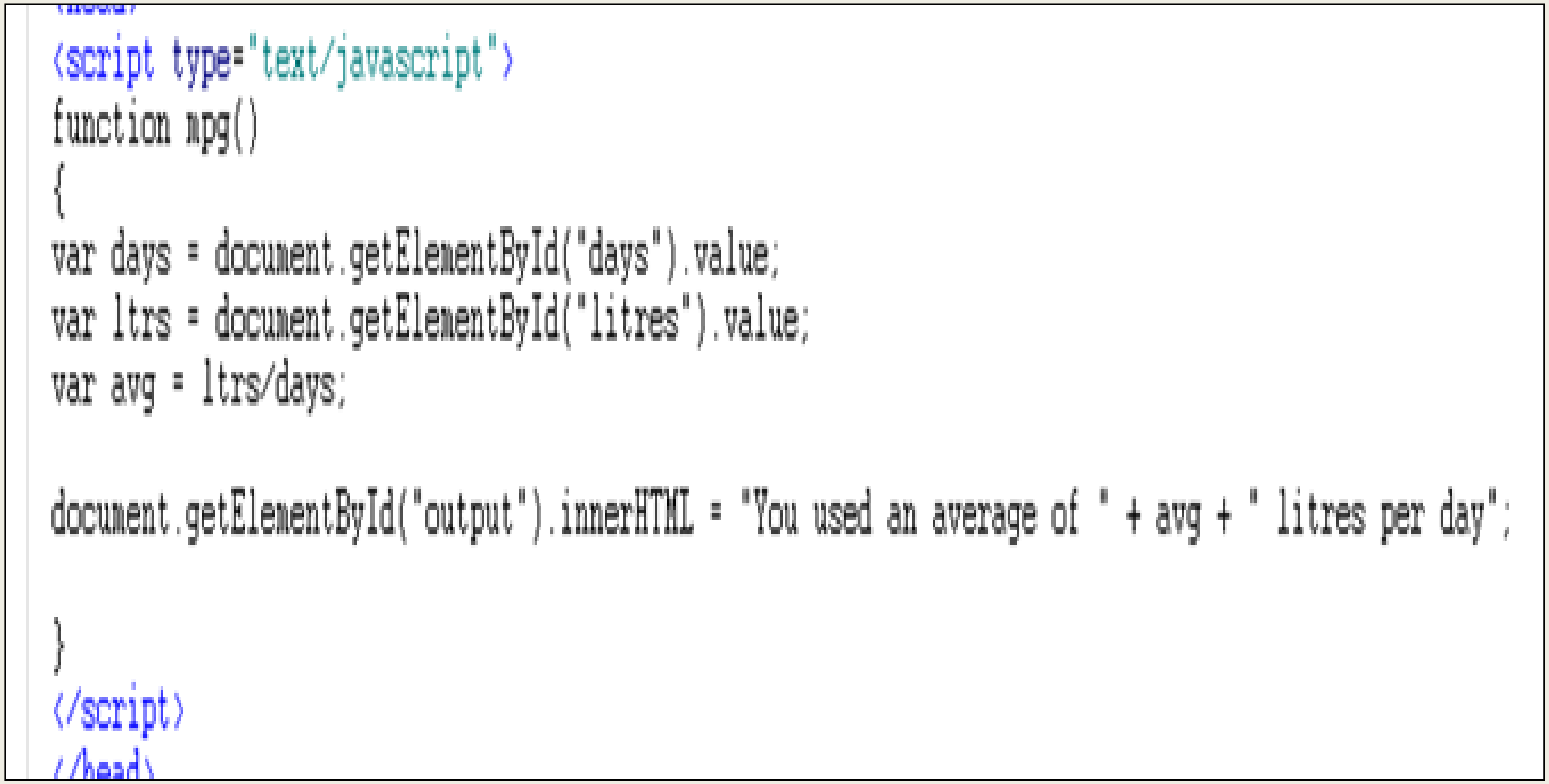
HTML - using ids



```
<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()" />
```

JavaScript

WaterUsage.html

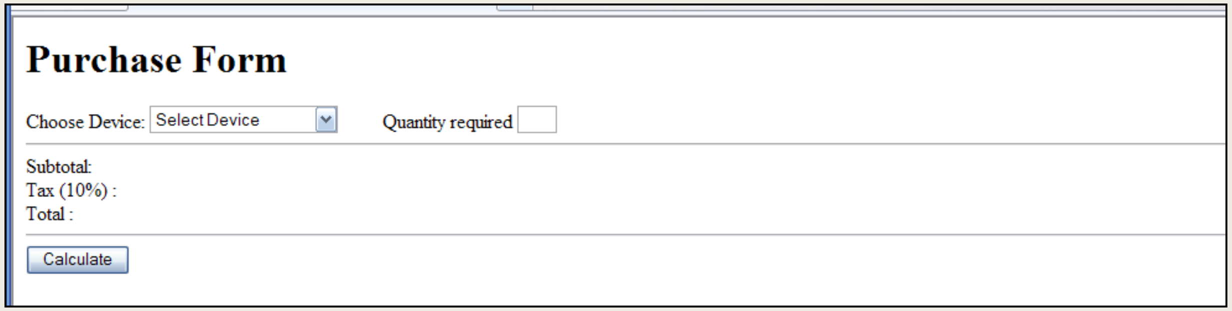


```
<script type="text/javascript">
function mpg()
{
var days = document.getElementById("days").value;
var ltrs = document.getElementById("litres").value;
var avg = ltrs/days;

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

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

Calculation Question 2

- Create a Purchase Form.
- The user selects a device from a dropdown menu
- The user enters the quantity required and presses submit.
- The subtotal (item price * quantity), tax and grandtotal are displayed.
- A popUp message is also displayed.

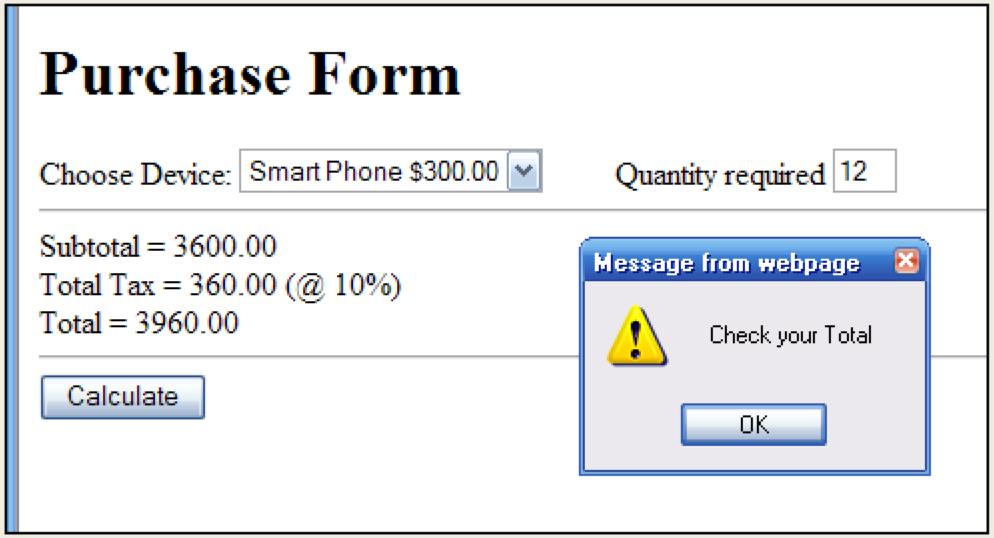
Calculations using JavaScript



Purchase Form

Choose Device: Select Device Quantity required

Subtotal:
Tax (10%) :
Total :



Purchase Form

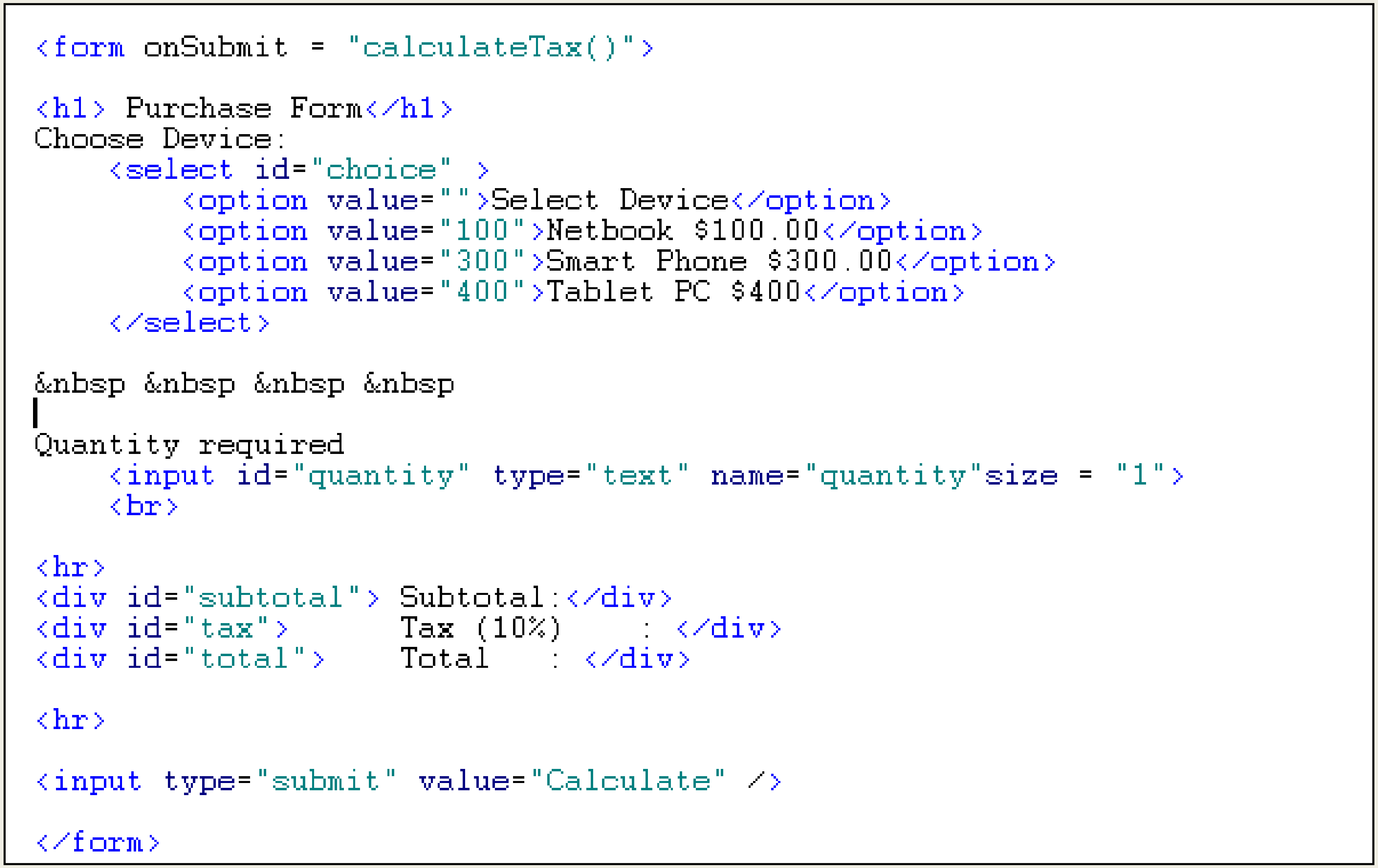
Choose Device: Smart Phone $300.00 Quantity required 12

Subtotal = 3600.00
Total Tax = 360.00 (@ 10%)
Total = 3960.00

Message from webpage

Check your Total

Calculation Question 2 – HTML

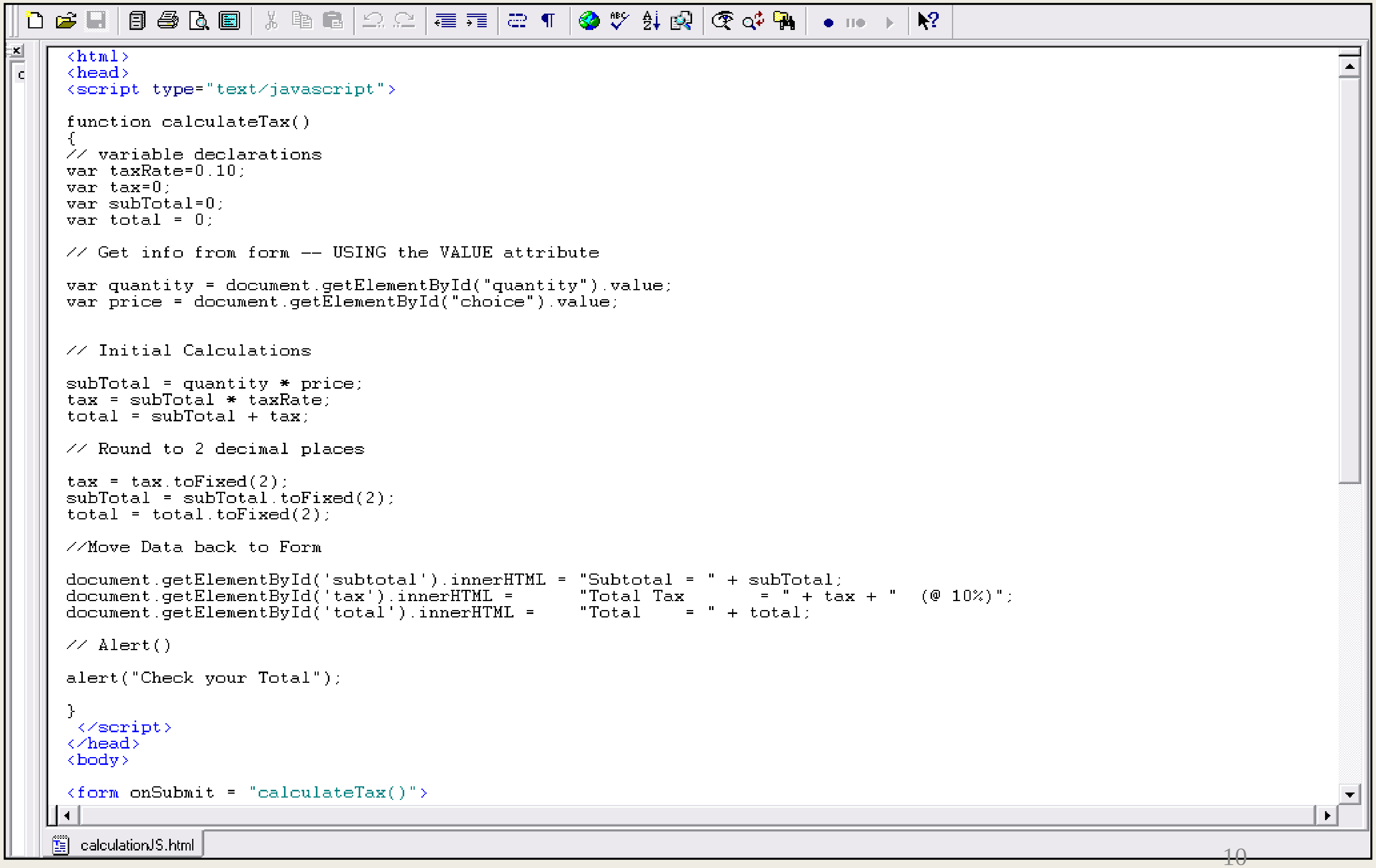


```
<form onSubmit = "calculateTax()">

<h1> Purchase Form</h1>
Choose Device:
    <select id="choice" >
        <option value="">Select Device</option>
        <option value="100">Netbook $100.00</option>
        <option value="300">Smart Phone $300.00</option>
        <option value="400">Tablet PC $400</option>
    </select>

&nbsp &nbsp &nbsp &nbsp

Quantity required
    <input id="quantity" type="text" name="quantity"size = "1">
    <br>

<hr>
<div id="subtotal"> Subtotal:</div>
<div id="tax">      Tax (10%)     : </div>
<div id="total">    Total    : </div>

<hr>

<input type="submit" value="Calculate" />

</form>
```

Calculation Question 2 – HTML

- Call function calculateTax() when press Submit button
- Using onSubmit() eventhandler placed in the <form> tag
- Notice – the price of each item is stored in the value attribute of the <option> tag
- E.g.
- <option value = “100”>Netbook $100</option>
- This enables you to use the price of the item in a calculation in JavaScript.

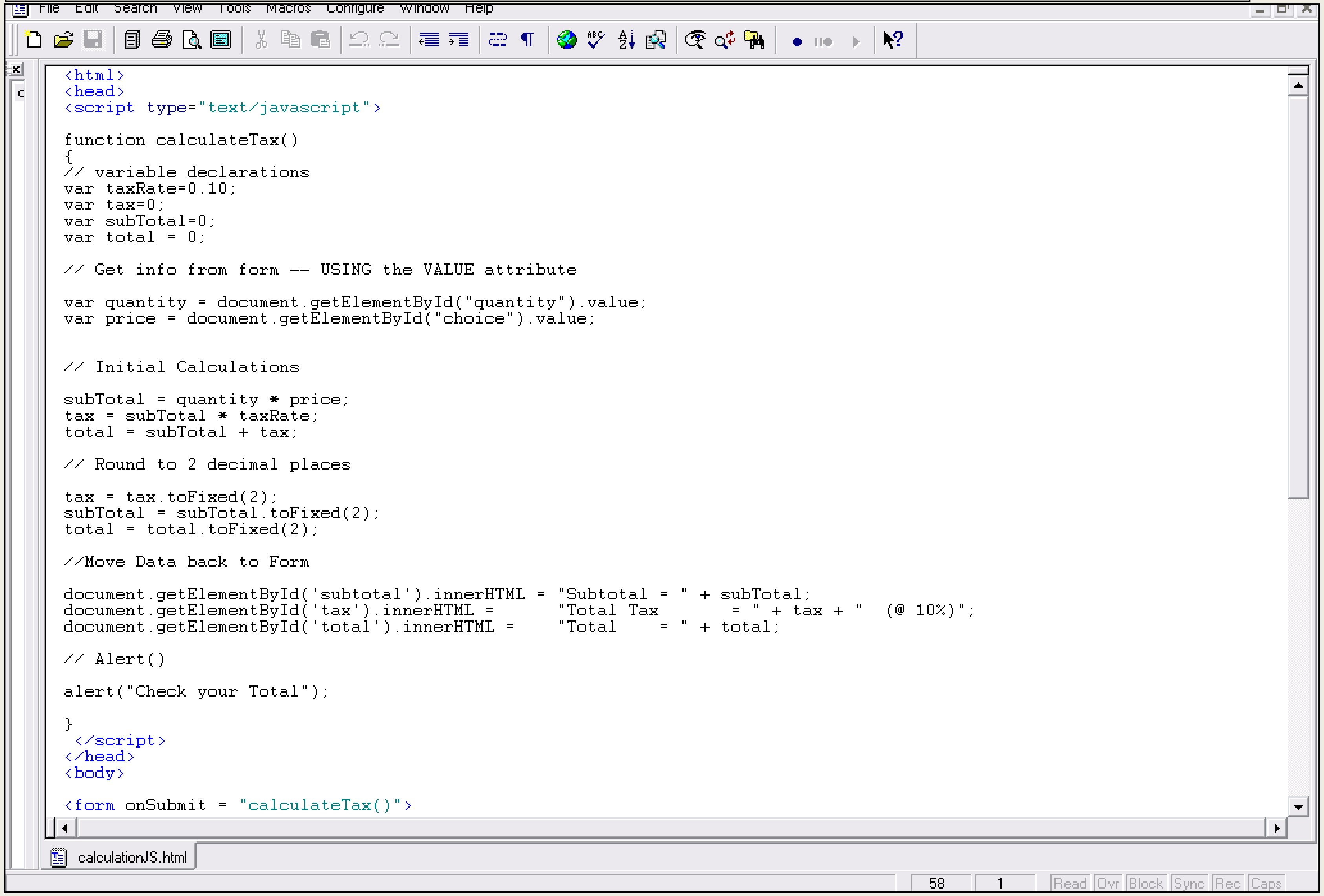
Calculations using JavaScript ..1



```
<html>
<head>
<script type="text/javascript">

function calculateTax()
{
// variable declarations
var taxRate=0.10;
var tax=0;
var subTotal=0;
var total = 0;

// Get info from form -- USING the VALUE attribute

var quantity = document.getElementById("quantity").value;
var price = document.getElementById("choice").value;


// Initial Calculations

subTotal = quantity * price;
tax = subTotal * taxRate;
total = subTotal + tax;

// Round to 2 decimal places

tax = tax.toFixed(2);
subTotal = subTotal.toFixed(2);
total = total.toFixed(2);

//Move Data back to Form

document.getElementById('subtotal').innerHTML = "Subtotal = " + subTotal;
document.getElementById('tax').innerHTML =      "Total Tax       = " + tax + "   (@ 10%)";
document.getElementById('total').innerHTML =    "Total     = " + total;

// Alert()

alert("Check your Total");

}
 </script>
</head>
<body>

<form onSubmit = "calculateTax()">
```

calculationJS.html

JavaScript..2



File Edit Search View Tools Macros Configure Window Help
calculationJS.html
58
1
Read Ovr Block Sync Rec Caps

```
<html>
<head>
<script type="text/javascript">

function calculateTax()
{
// variable declarations
var taxRate=0.10;
var tax=0;
var subTotal=0;
var total = 0;

// Get info from form -- USING the VALUE attribute

var quantity = document.getElementById("quantity").value;
var price = document.getElementById("choice").value;


// Initial Calculations

subTotal = quantity * price;
tax = subTotal * taxRate;
total = subTotal + tax;

// Round to 2 decimal places

tax = tax.toFixed(2);
subTotal = subTotal.toFixed(2);
total = total.toFixed(2);

//Move Data back to Form

document.getElementById('subtotal').innerHTML = "Subtotal = " + subTotal;
document.getElementById('tax').innerHTML =      "Total Tax        = " + tax + "   (@ 10%)";
document.getElementById('total').innerHTML =    "Total     = " + total;

// Alert()

alert("Check your Total");

}
 </script>
</head>
<body>

<form onSubmit = "calculateTax()">
```

Calculations using JavaScript

- Declare variables for using in program
- Get data from Form – using the value attribute with getElementById("")
- Do the initial calculations – subtotal, tax, grand total
- Round the figures to 2 decimal places, using .toFixed(2)
- Move the data back into the form using document.getElementById(" ").innerHTML =

ONSUBMIT()

Displaying results of calculations

On Submit() function

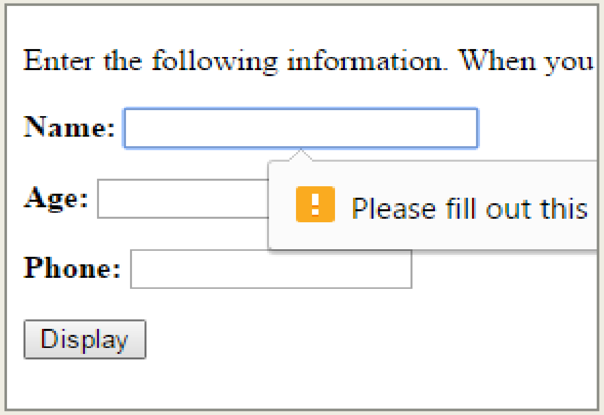
Form – OnSubmit

- Put onSubmit() in form tag always
- Calls the function

```
<body>

<form action="#" onsubmit="display();">

<p><strong>Name:</strong>

<input type ="text" size="20" name ="yourname" id = "name" required>

</p>

<p><input type = "submit" value="Display"></p>

</form>

</body>

</html>
```

Call function display()

```
<script language="JavaScript" type="text/javascript">
function display() {
  var name = document.getElementById("name").value;
alert("Name: " + name + "\n Phone: " + phone + "\n Age: " + age);
}
</script>
```

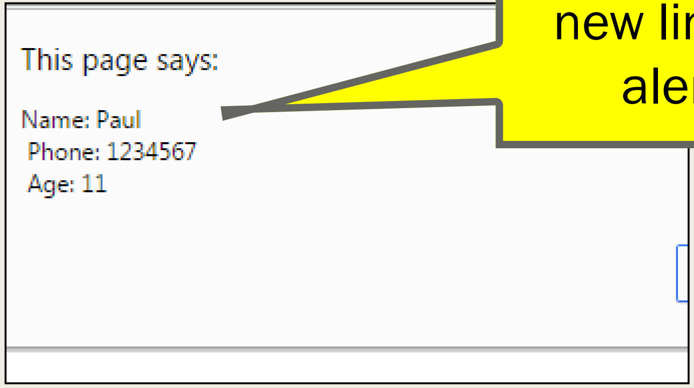

Enter the following information. When you

Name:

Age:

Please fill out this

Phone:


This page says:

```
Name: Paul
 Phone: 1234567
 Age: 11
```

Use \n for new line in alert

EVENT HANDLERS

Rollover

Random Tip display

Image Gallery

Hide / Show / Toggle

Order Form

Calculations

Event handlers

EventHandler	Called when . . .
onClick	User clicks on page 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

- Can create dynamic effects using Event Handlers
- Can respond to user actions
- Can change the appearance of an object in a HTML document using JavaScript
- Use DOM presentational properties (using style)
- These have similar names to CSS stylesheet properties except hyphenated CSS properties adopt camel case for DOM property names
- E.g. text-align becomes textAlign

Event Handlers

- CSS text-align property becomes textAlign in DOM property
- All DOM presentational properties are contained in the style property
- Use dot notation

```
var head = document.getElementById(“header”);
 head.style. border = “1px solid red”;
 head.style. background = “yellow”;
 head.style. textAlign = “center”;
```

More event Examples

- 1. Rollovers
- 2. Toggle visibility
- 3. Random Tip
- 4. Slideshow
- 5. Order Form

Rollovers

- Rollovers use the following events:
  - ***onMouseOver***
  - *onMouseOut*
- Browsers deal with the position (x,y) in different ways

Rollover

- Attach the **OnMouseOver** and **OnMouseOut** to images
- These call two functions – **swapOut**() and **swapBack()**

```
<img src=“images/kitten.jpg" id="kitty"
      onmouseover = "swapOut();"
        onmouseout   =  "swapBack();“ >
```

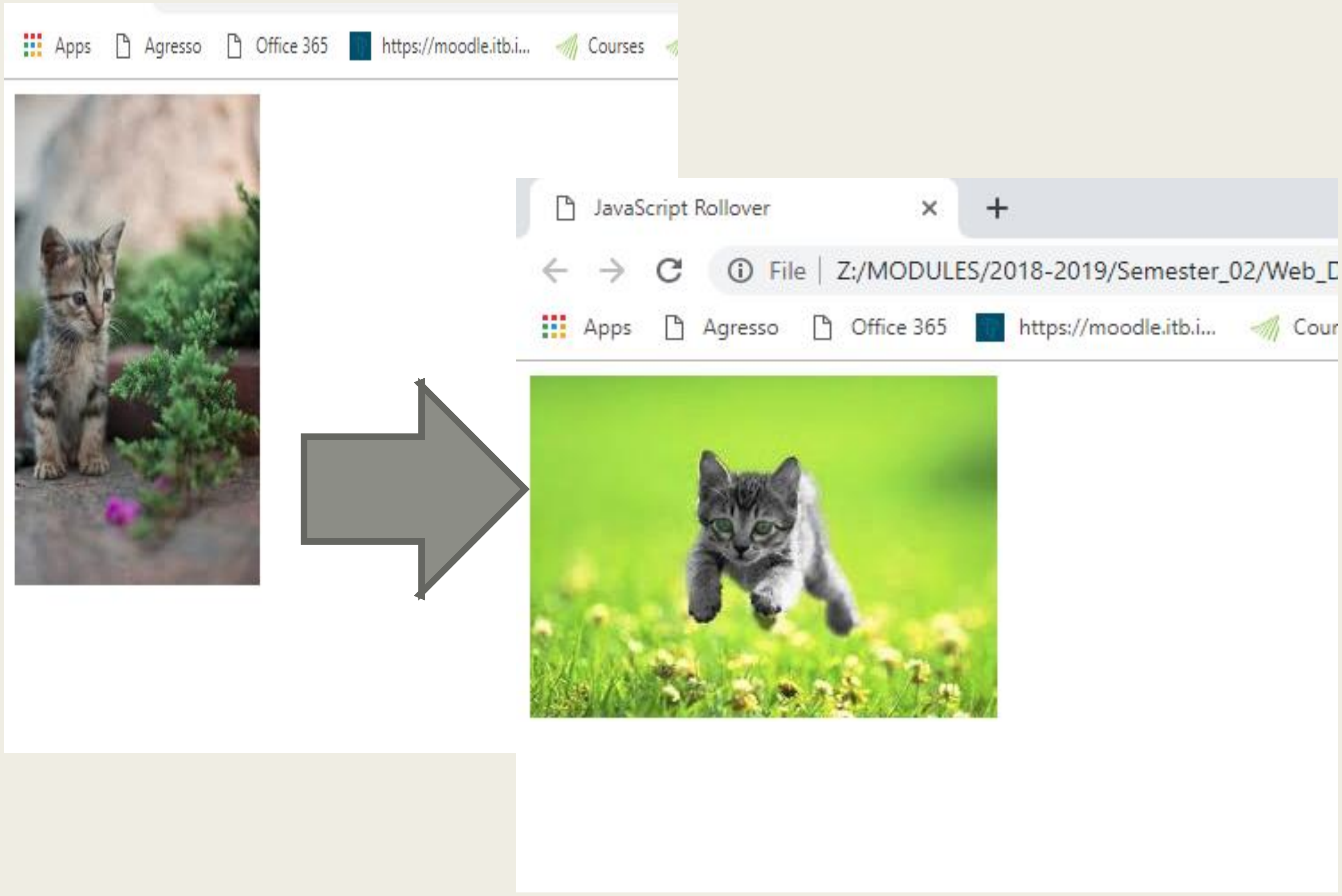
Rollover – JavaScript ..1

```
<script type = "text/javascript">
 // Declare image1 + image size
          var rollimage1= new Image(100,200);
 // image2 address
           rollimage1.src = “images/kitten.jpg";
 // Declare image2 + size
          var rollimage2 = new Image(100,200);
 // image2 address
          rollimage2.src = “images/kitten2.jpg";
```

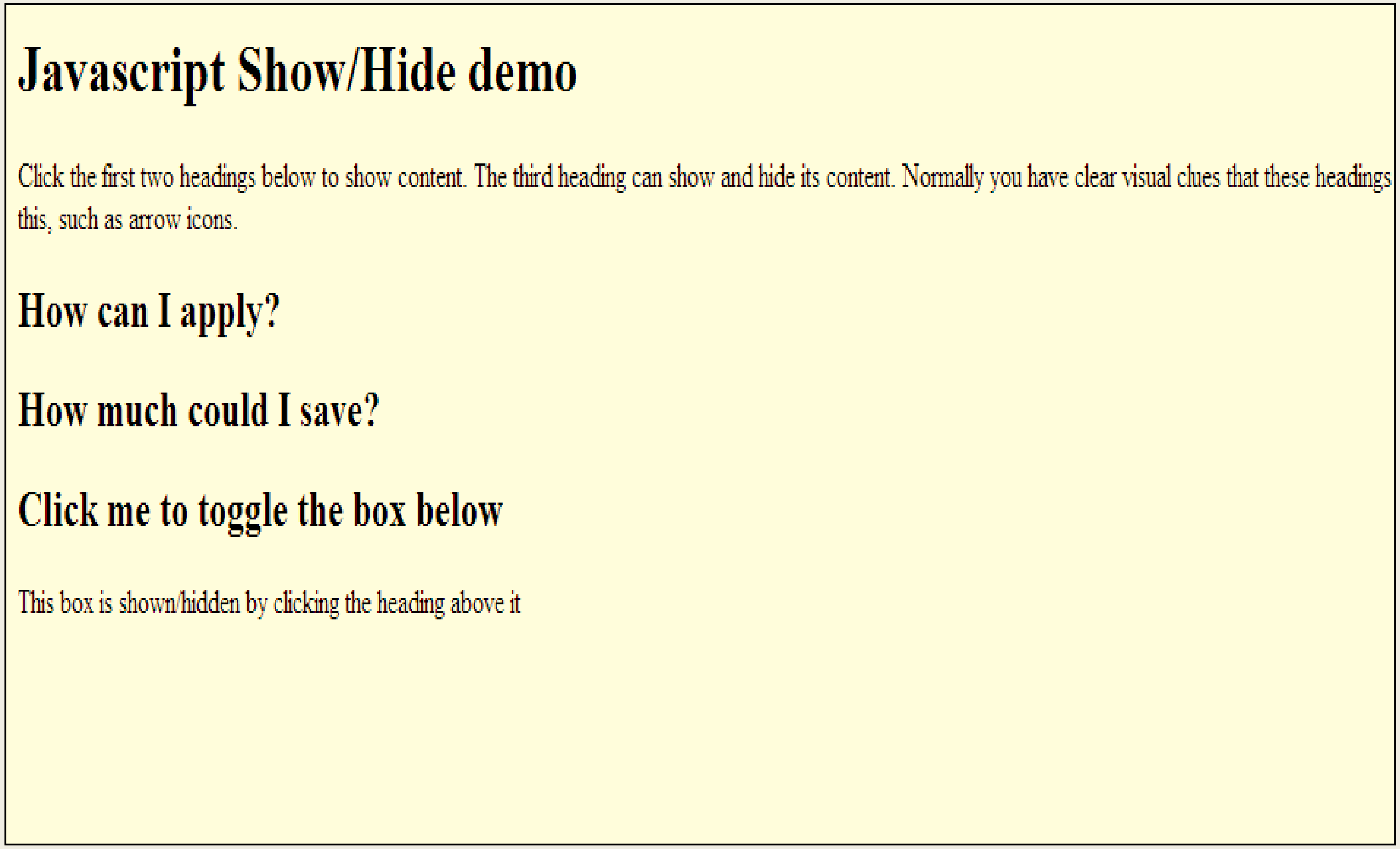
Rollover – JavaScript ..2

```
// rollover functions
 function swapOut(){
   document.getElementById("kitty").src =
   rollimage1.src;
 }
 function swapBack(){
   document.getElementById("kitty").src =
   rollimage2.src;
 }
```

Screenshot


Apps
Agresso
Office 365
https://moodle.itb.i...
Courses
JavaScript Rollover
File
Z:/MODULES/2018-2019/Semester_02/Web_D
Apps
Agresso
Office 365
https://moodle.itb.i...
Cour

Show / Hide / Toggle a div



Javascript Show/Hide demo

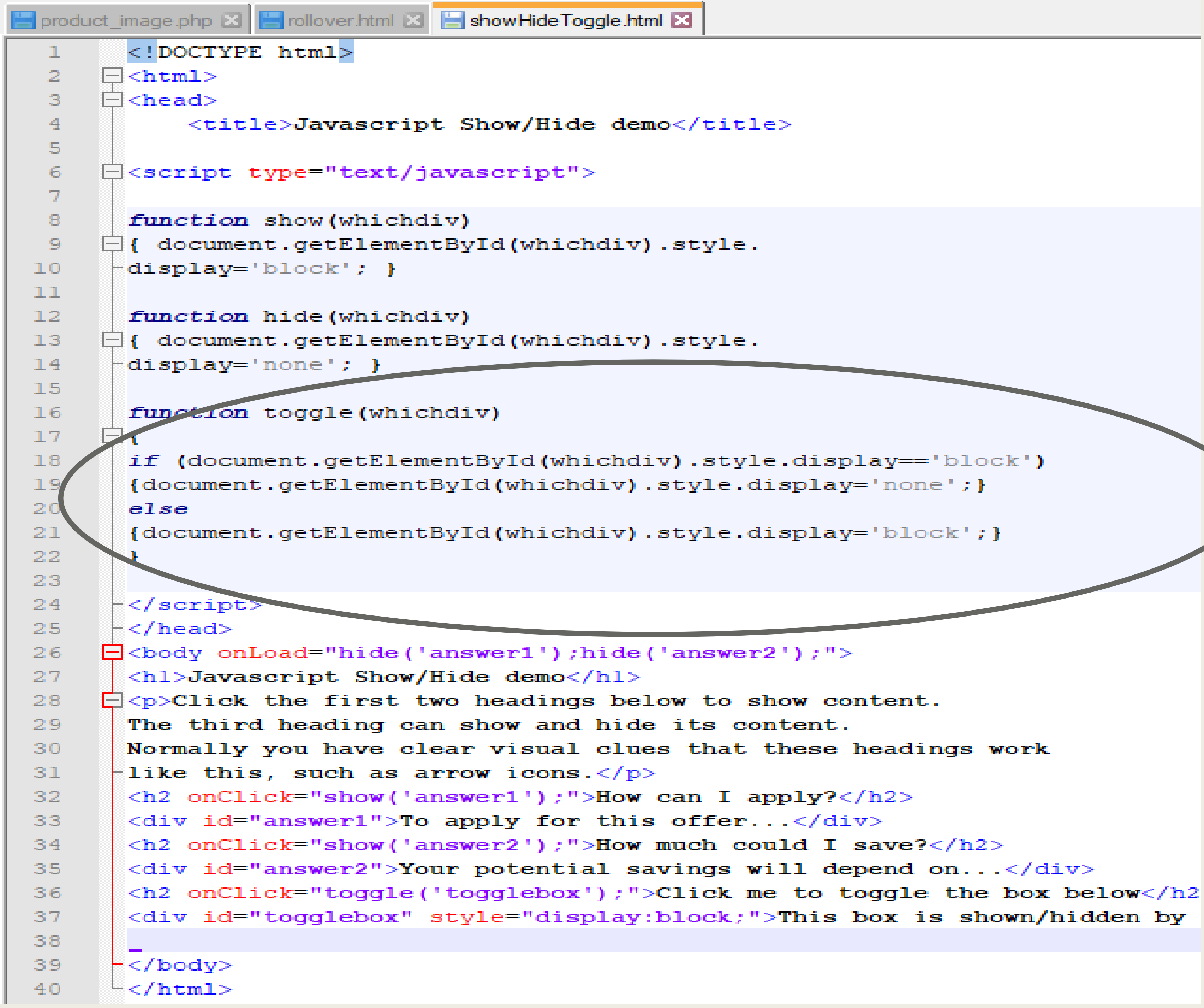
Click the first two headings below to show content. The third heading can show and hide its content. Normally you have clear visual clues that these headings
this, such as arrow icons.

How can I apply?

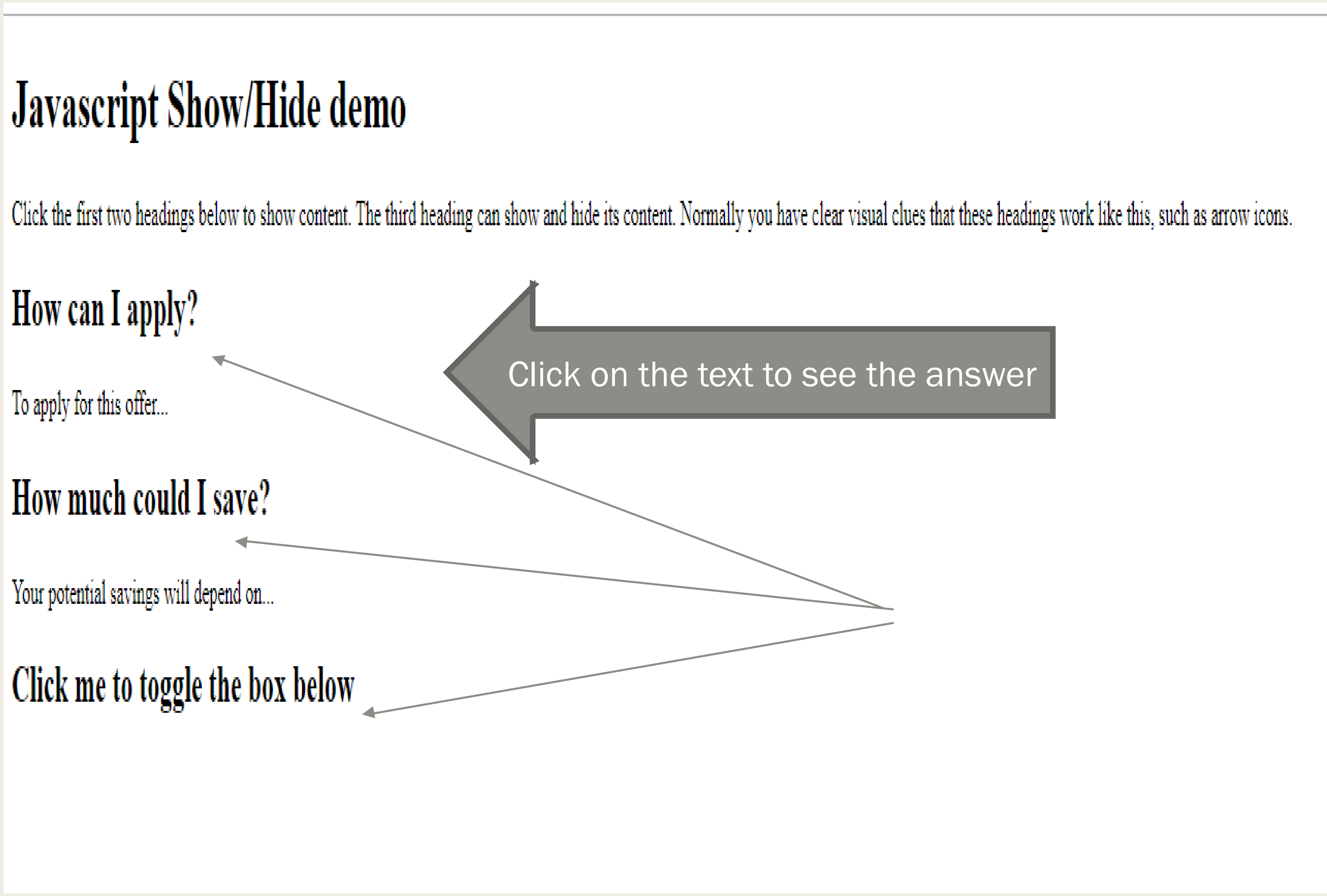
How much could I save?

Click me to toggle the box below

This box is shown/hidden by clicking the heading above it



```
<!DOCTYPE html>
<html>
<head>
    <title>Javascript Show/Hide demo</title>

<script type="text/javascript">

function show(whichdiv)
{ document.getElementById(whichdiv).style.
display='block'; }

function hide(whichdiv)
{ document.getElementById(whichdiv).style.
display='none'; }

function toggle(whichdiv)
{
if (document.getElementById(whichdiv).style.display=='block')
{document.getElementById(whichdiv).style.display='none';}
else
{document.getElementById(whichdiv).style.display='block';}
}

</script>
</head>
<body onLoad="hide('answer1');hide('answer2');">
<h1>Javascript Show/Hide demo</h1>
<p>Click the first two headings below to show content.
The third heading can show and hide its content.
Normally you have clear visual clues that these headings work
like this, such as arrow icons.</p>
<h2 onClick="show('answer1');">How can I apply?</h2>
<div id="answer1">To apply for this offer...</div>
<h2 onClick="show('answer2');">How much could I save?</h2>
<div id="answer2">Your potential savings will depend on...</div>
<h2 onClick="toggle('togglebox');">Click me to toggle the box below</h2
<div id="togglebox" style="display:block;">This box is shown/hidden by

</body>
</html>
```



Javascript Show/Hide demo

Click the first two headings below to show content. The third heading can show and hide its content. Normally you have clear visual clues that these headings work like this, such as arrow icons.

How can I apply?

To apply for this offer...

How much could I save?

Your potential savings will depend on...

Click me to toggle the box below

Click on the text to see the answer

Using CSS – Display property

- Display:block
- Display:inline
- Display: none

Show / Hide / Toggle a div

```
<body onLoad="hide('answer1');hide('answer2');">


<h2 onClick="show('answer1');">How can I apply?</h2>
    <div id="answer1">To apply for this offer...</div>
<h2 onClick="show('answer2');">How much could I
    save?</h2>
    <div id="answer2">Your savings will depend on...</div>


<h2 onClick="toggle('togglebox');">Click me to toggle the
    box below</h2>
<div id="togglebox" style="display:block;">This box is
    shown/hidden by clicking the heading above it</div>
```

Hide the contents – visibility.html

- When the page is loaded, the onLoad event handler is called.
- This event handler calls on function hide() twice with 2 different parameters:
  - *hide('answer1') and hide('answer2')*

```
function hide(whichdiv)
{ document.getElementById(whichdiv).style. display='none'; }
```

Show the contents

- When the user clicks on either of the first two questions, the onClick event handler calls the show() function and an appropriate parameter is passed i.e. answer1 or answer2.

```
function show(whichdiv)
{
document.getElementById(whichdiv).style.
display='block'; }
```

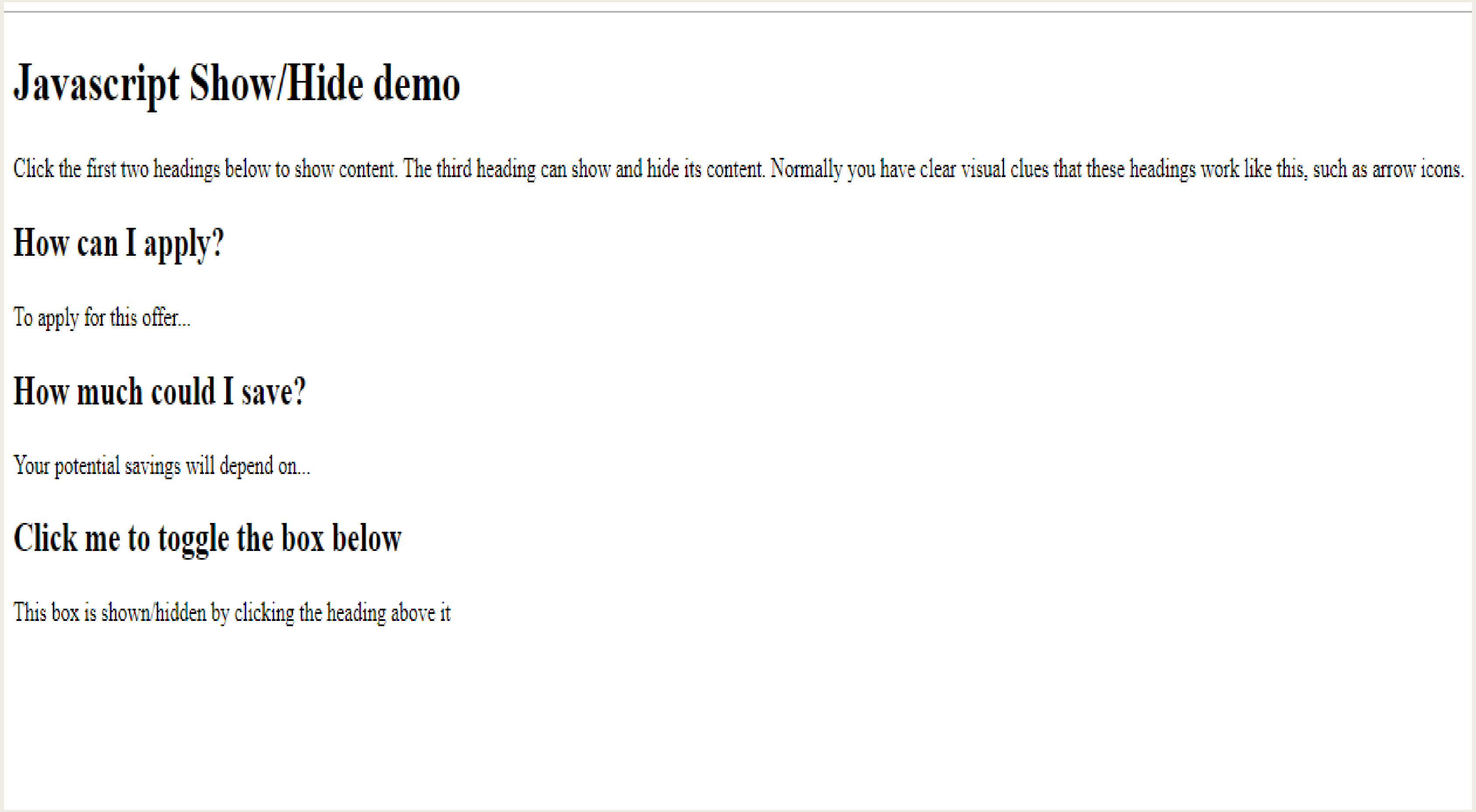
Toggle the contents

- When the user clicks the third heading – the contents with be revealed or will disappear, on every second user click (i.e. toggle)
- The function toggle() is called and it checks to see if the block of text is displayed or hidden
- If …..(style. display == ‘block’) i.e. visible, then make it invisible, by changing (style. display = ‘none’)
- If nothing is on display, then make the text visible (style. display = ‘block’)

Show / Hide / Toggle ...js

```
function toggle(whichdiv){
if  (document.getElementById(whichdiv).style.display= ='block')
{
    document.getElementById(whichdiv).style. display = 'none';
}
else
{
    document.getElementById(whichdiv).style.display='block';}
}
```

screenshot



Javascript Show/Hide demo

Click the first two headings below to show content. The third heading can show and hide its content. Normally you have clear visual clues that these headings work like this, such as arrow icons.

How can I apply?

To apply for this offer...

How much could I save?

Your potential savings will depend on...

Click me to toggle the box below

This box is shown/hidden by clicking the heading above it

Toggle visibility

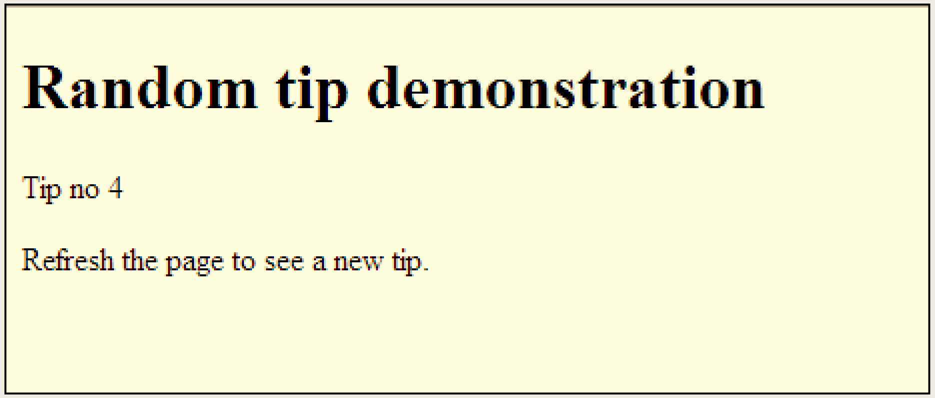
- Previous program could have used the CSS visibility property
- CSS:
  - *visibility: visible;*
  - *visibility: hidden*
- In JavaScript this would be written as:
  - *xxxx.style.visibility = “visible”;*
  - *xxxx.style.visibility = “hidden”;*

Display – V- Visibility

- The visibility property specifies whether or not an element is visible.
- Even invisible elements takes up space on the page.
- Use the "display" property to create invisible elements that do not take up space!

Random Tip Demo … example of arrays

```
<body onLoad="tip_setup();" >
<h1>Random tip demonstration</h1>
    <div id="tipbox"></div>
    <p>Refresh the page to see a new tip.</p>
```



Random tip demonstration

Tip no 4

Refresh the page to see a new tip.

Random Tip

- Allocate space for an array and assign it to the variable tips.

  tips = new Array();

- Fill each array index with some text
- Use Math.random() to randomly chose a tip to display
- The array in JavaScript is similar to an array in Java, but you don't have to declare its length.
- Find out its lengths using ….*arrayName.length*

Random Tip Demo ... js

```
function tip_setup() {
tips=new Array();
tips[0] ="<strong>First tip goes here</strong>";
tips[1] ="Second tip here";
tips[2] ="Third tip here";
tips[3] = "Tip no 4";
var chosenOne = Math.floor(Math.random()* tips.length);
//display the tip
document.getElementById('tipbox').innerHTML=tips[chosen
    One];
}
```

Tips code

```
Math.floor(Math.random()* tips.length);
```

- tips.length --- finds the length of the tips array
- Math.randon() – finds a random number
- Math.floor() – rounds a number downwards to nearest integer
- tips[chosenOne] ---selects an element in the array

Photo Slideshow … html

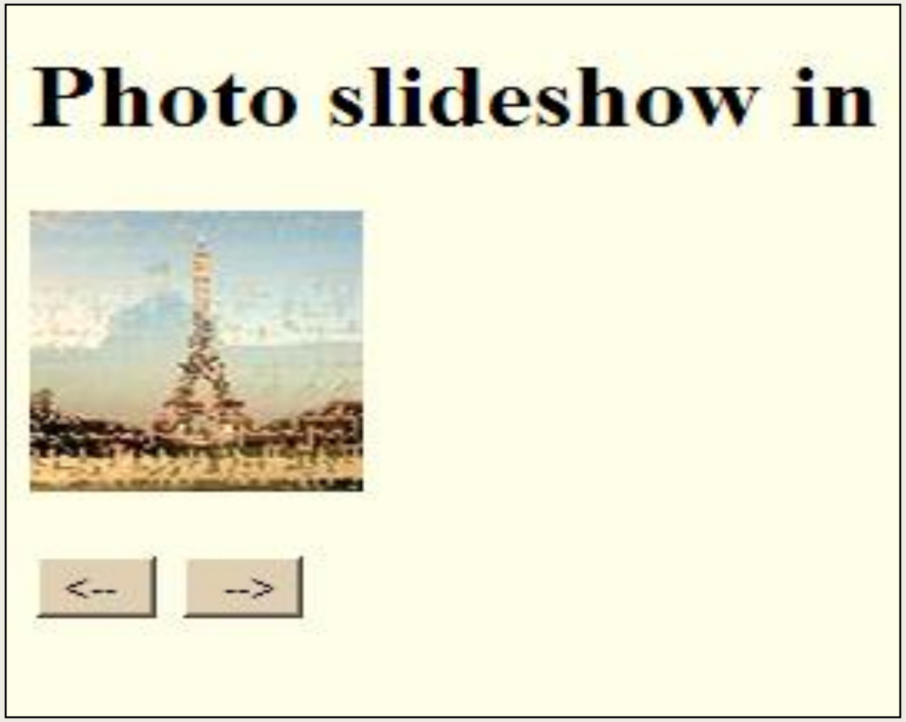


Photo slideshow in

Photo Slideshow ... html

```
<form>
<img src="slidepics/1.jpg"  id="slideshow" alt="Photos
    of Paris" width = 100; height = "100">
<p>
<input type="button" value="<-- " onClick="newSlide(-
    1)">
<input type="button" value=" -->"
    onClick="newSlide(1)">
</p>
</form>
```

Photslideshow ...js

```
thisImg = 1;   // assign 1 to first image
imgCt = 17;   // assign 17 to last image


function newSlide(direction) {
          thisImg = thisImg + direction;
            if (thisImg < 1) {
                      thisImg = imgCt; }
           if (thisImg == imgCt+1) {
                      thisImg = 1;  }
document.getElementById('slideshow').src = "slidepics/" +
     thisImg + ".jpg";
     }
```

Photo Slideshow

- Images stored in folder ..slideshow
- Images called .. 1.jpg, 2.jpg ….3.jpg
- Src of images is located using value in thisImg
  - *slidepics/" + thisImg + ".jpg";*
- If you want to go forward, press the forward arrow which calls function newSlide() and passes it the value 1.
- Direction is equal to 1 then.
- Add this value to variable thisIMg

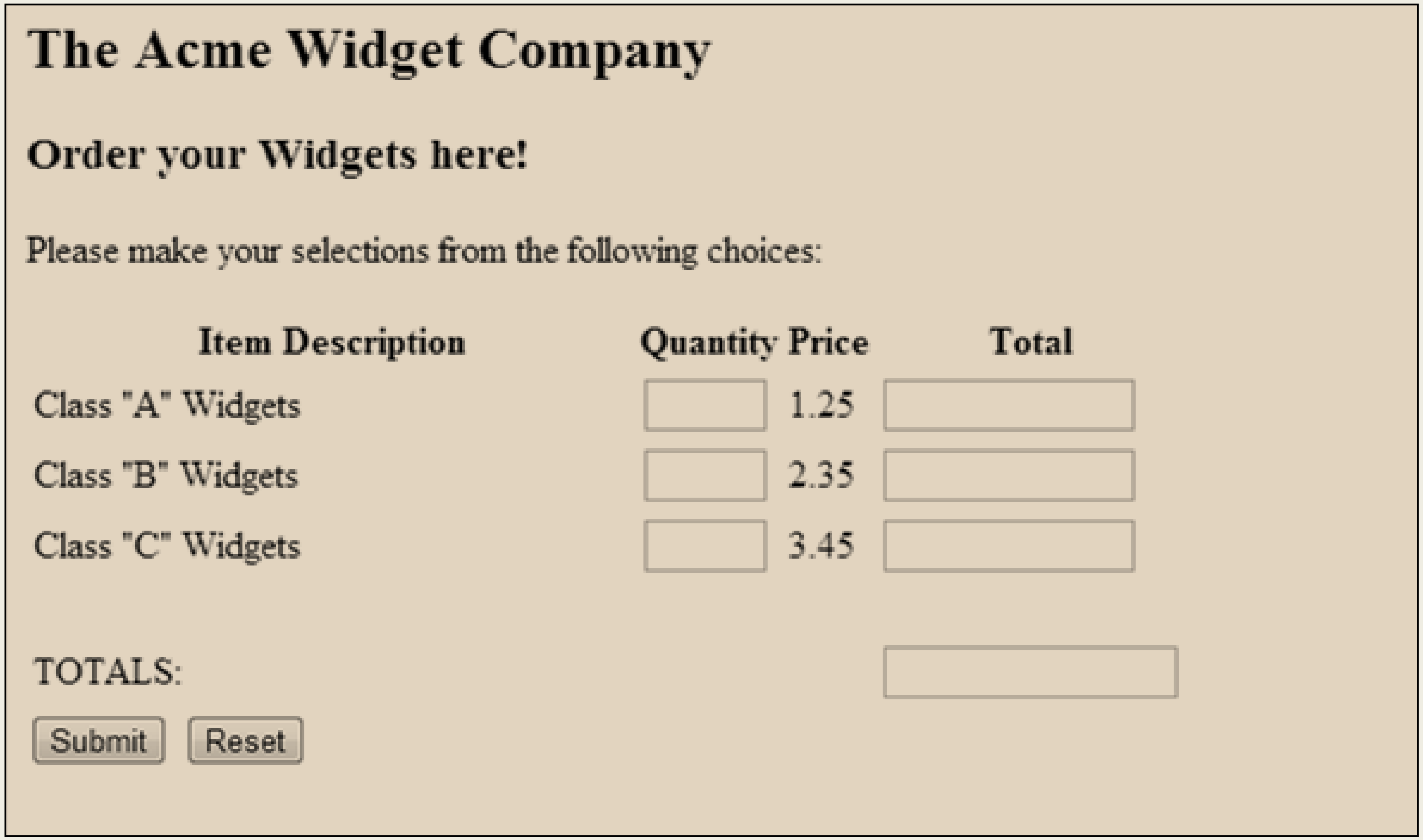
Photo Slideshow

- If the value of thisImg is less than one, then display the last image (no. 17)
- If the value of thisImg is equal to 18, then change the value of thisImg to 1.
- If the user decides to go backwards from a particular slide, then the function newSlide(-1) has a value of minus one (-1). This is sent to the function and 1 is deducted from the current slide number and the previous slide is displayed.
- In this way, you can go forward and backwards through the slide set.

ORDER FORM

OnChange event handler

Order Form



The Acme Widget Company

Order your Widgets here!

Please make your selections from the following choices:

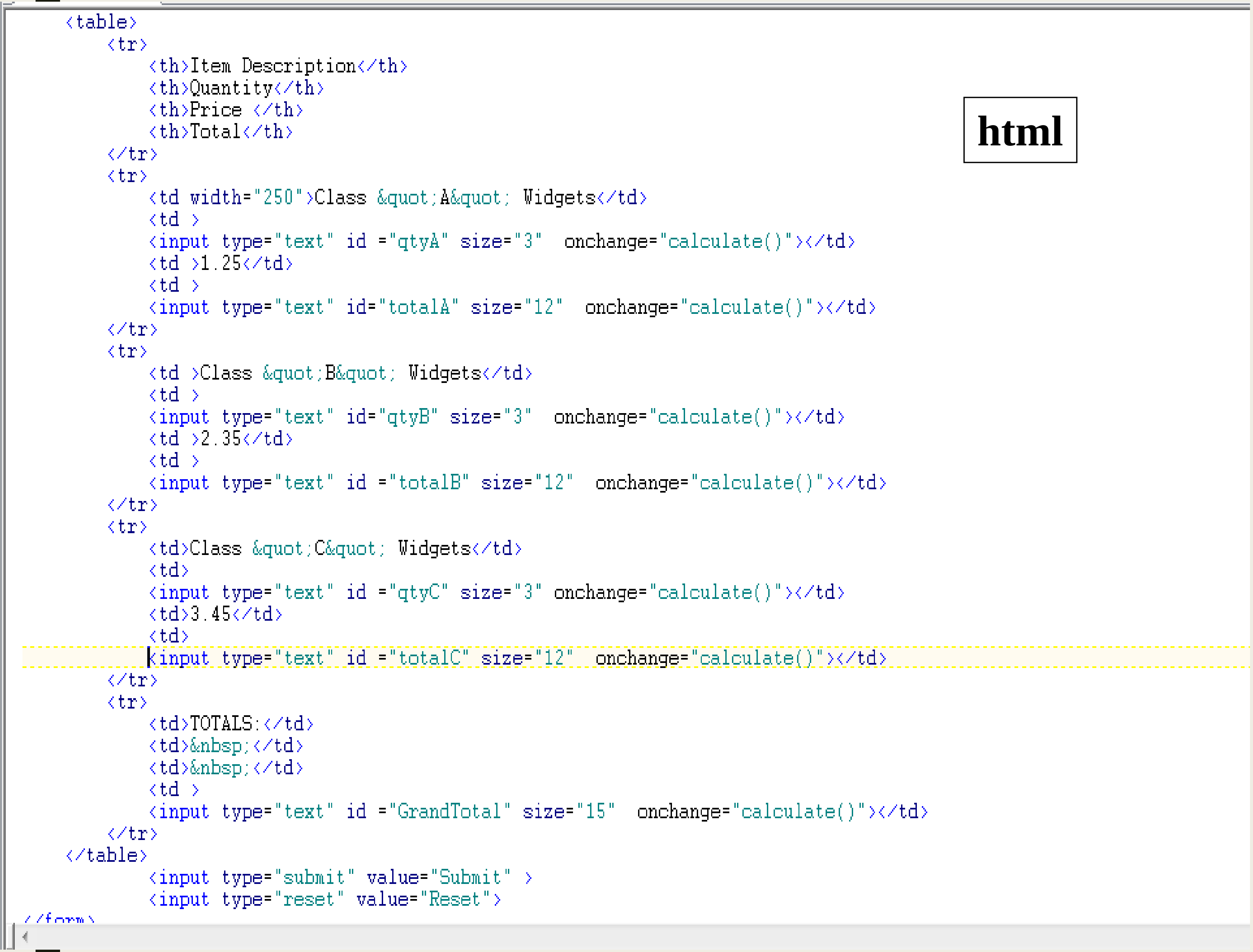
Item Description	Quantity	Price	Total
Class "A" Widgets		1.25	
Class "B" Widgets		2.35	
Class "C" Widgets		3.45	

TOTALS:

Tables -recap

<u>TABLE - recap</u>

```
<table>
     <tr>
          <th> .......</th> <th>.........</th>
   </tr>
      <tr>
          <td> .......</td><td>.........</td>
  </tr>
     <tr><td> .......</td><td>.........</td></tr>
  </table>
```



html

```
    <table>
        <tr>
            <th>Item Description</th>
            <th>Quantity</th>
            <th>Price </th>
            <th>Total</th>
        </tr>
        <tr>
            <td width="250">Class &quot;A&quot; Widgets</td>
            <td >
            <input type="text" id ="qtyA" size="3"  onchange="calculate()"></td>
            <td >1.25</td>
            <td >
            <input type="text" id="totalA" size="12"  onchange="calculate()"></td>
        </tr>
        <tr>
            <td >Class &quot;B&quot; Widgets</td>
            <td >
            <input type="text" id="qtyB" size="3"  onchange="calculate()"></td>
            <td >2.35</td>
            <td >
            <input type="text" id ="totalB" size="12"  onchange="calculate()"></td>
        </tr>
        <tr>
            <td>Class &quot;C&quot; Widgets</td>
            <td>
            <input type="text" id ="qtyC" size="3" onchange="calculate()"></td>
            <td>3.45</td>
            <td>
            <input type="text" id ="totalC" size="12"  onchange="calculate()"></td>
        </tr>
        <tr>
            <td>TOTALS:</td>
            <td>&nbsp;</td>
            <td>&nbsp;</td>
            <td >
            <input type="text" id ="GrandTotal" size="15"  onchange="calculate()"></td>
        </tr>
    </table>
            <input type="submit" value="Submit" >
            <input type="reset" value="Reset">
</form>
```

Order Form ...html

- Using tables for layout of the form
- Using onChange event handler to trigger JavaScript.
- When something changes in the Quantity and the GrandTotal fields the **onChange** event handler calls the JavaScript **calculate()** function which is in the <head> of the file.

Order Form …JavaScript

- Declare the variables that are used to take in values from the form
- Declare the prices for the various items
- ITEM A: Check whether the quantity field is blank or not
- If it is filled in, fetch the quantity the user entered for ITEM A and put it into a variable
- Calculate the total for ITEM A i.e. price * quantity

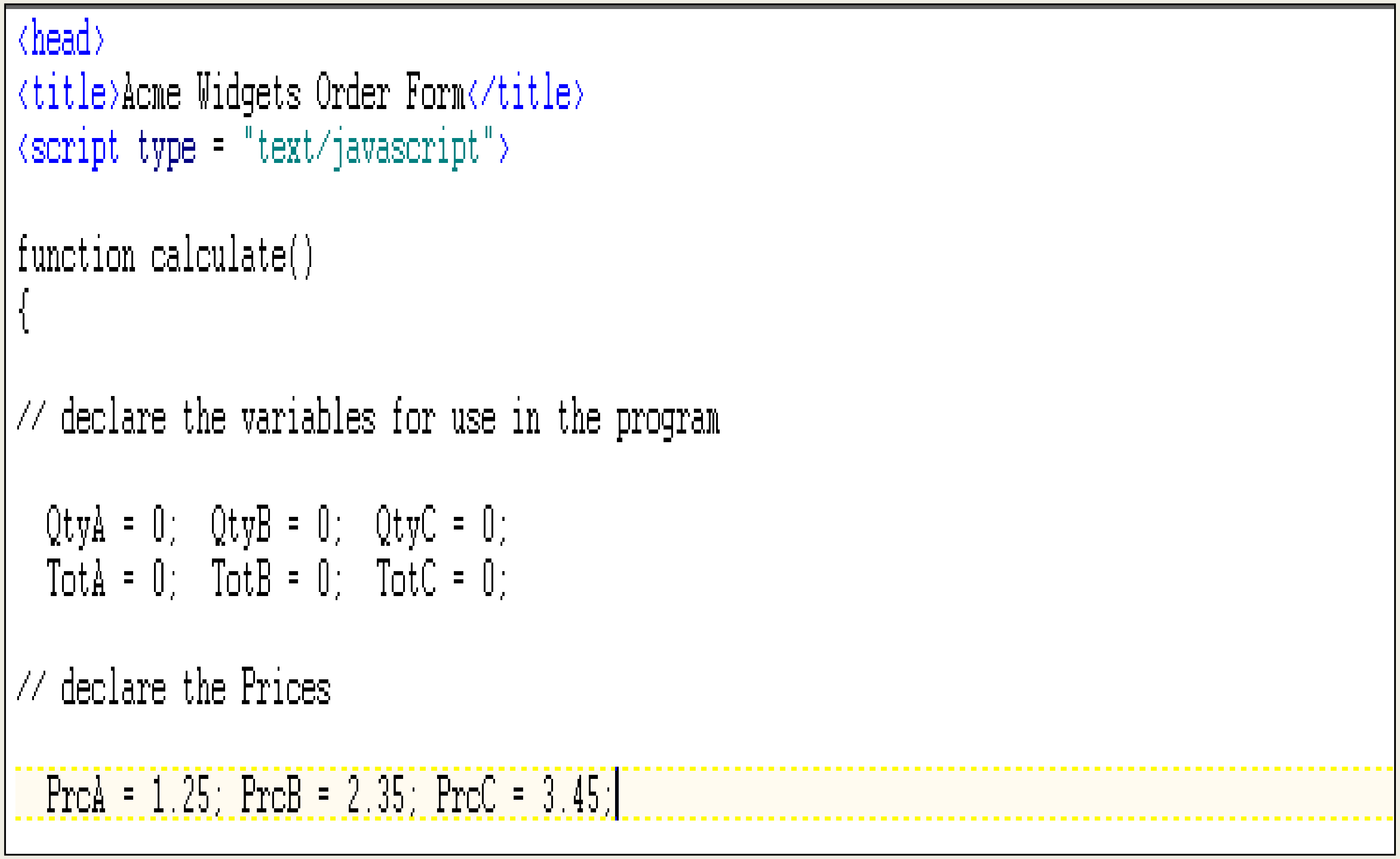
Order Form …JavaScript

- Use the eval() function for these calculations.
- eval() changes a text field into a number field
- Convert the total to 2 decimal places, using toFixed(2)
- Output the total for ITEM A
- Ditto for ITEM B
- Ditto for ITEM C
- Calculate the Grand Total for the order

Order Form ….JavaScript

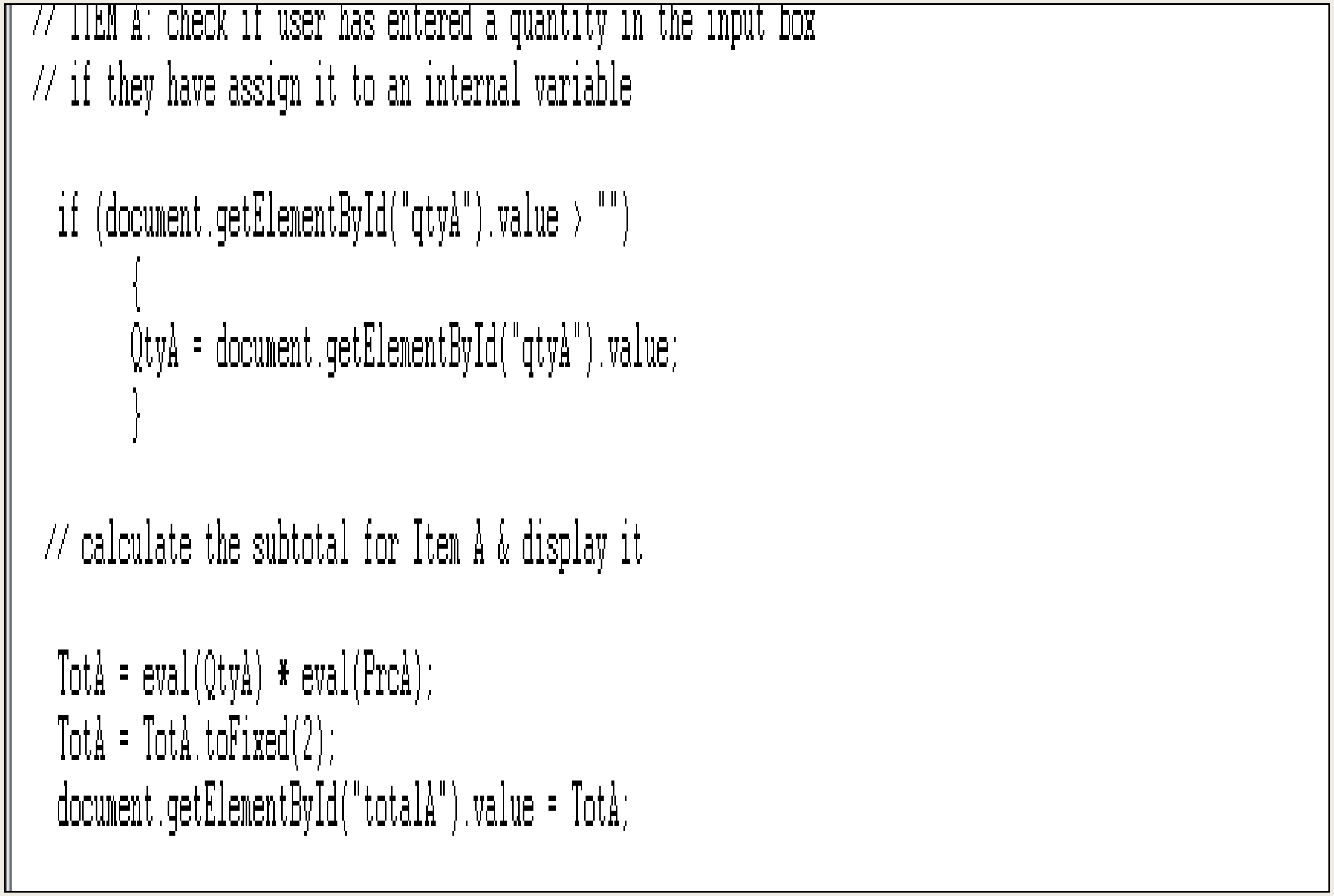
- Add the totals for Items A + B + C together for the Grand Total
- Use the eval() function to ensure that all values are converted to numbers.
- Convert the Grand Total to 2 decimal places and display it on the form.
- Note: The form shows a running total

JavaScript …..1



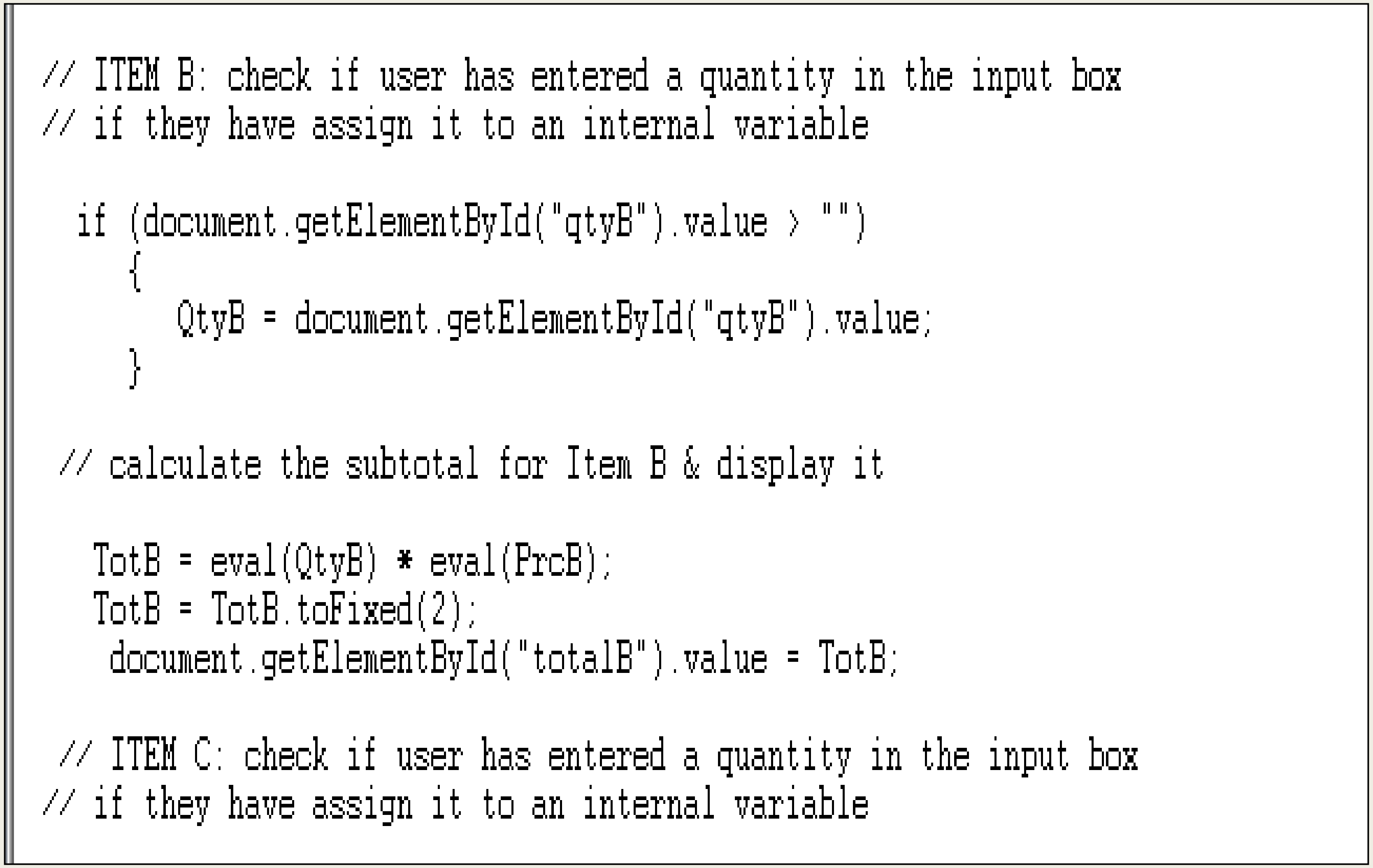
```
<head>
<title>Acme Widgets Order Form</title>
<script type = "text/javascript">

function calculate()
{

// declare the variables for use in the program

  QtyA = 0;  QtyB = 0;  QtyC = 0;
  TotA = 0;  TotB = 0;  TotC = 0;

// declare the Prices

  PrcA = 1.25; PrcB = 2.35; PrcC = 3.45;
```

JavaScript ….2



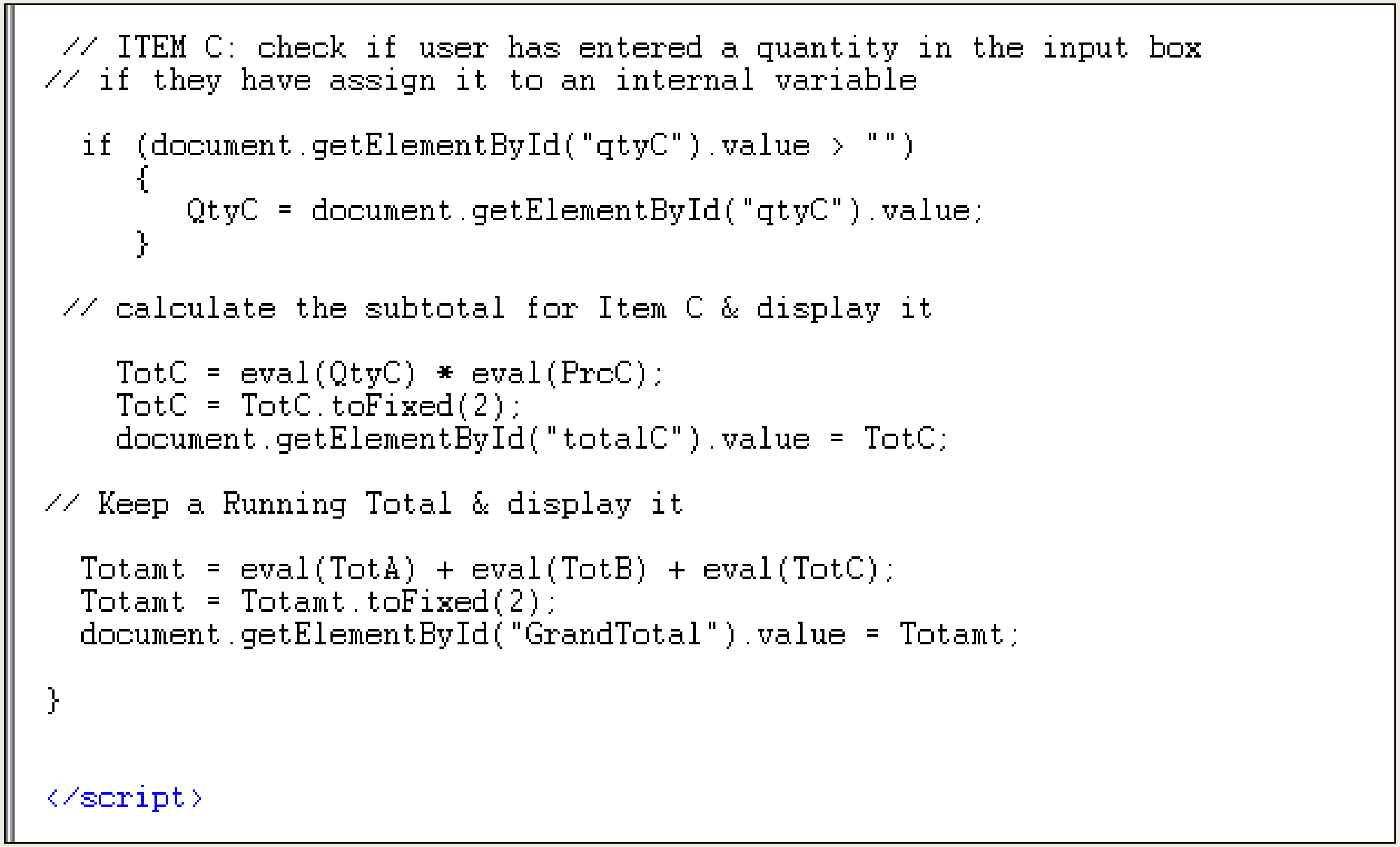
```
// ITEM A: check if user has entered a quantity in the input box
// if they have assign it to an internal variable

  if (document.getElementById("qtyA").value > "")
        {
        QtyA = document.getElementById("qtyA").value;
        }

 // calculate the subtotal for Item A & display it

  TotA = eval(QtyA) * eval(PrcA);
  TotA = TotA.toFixed(2);
  document.getElementById("totalA").value = TotA;
```

JavaScript ….3



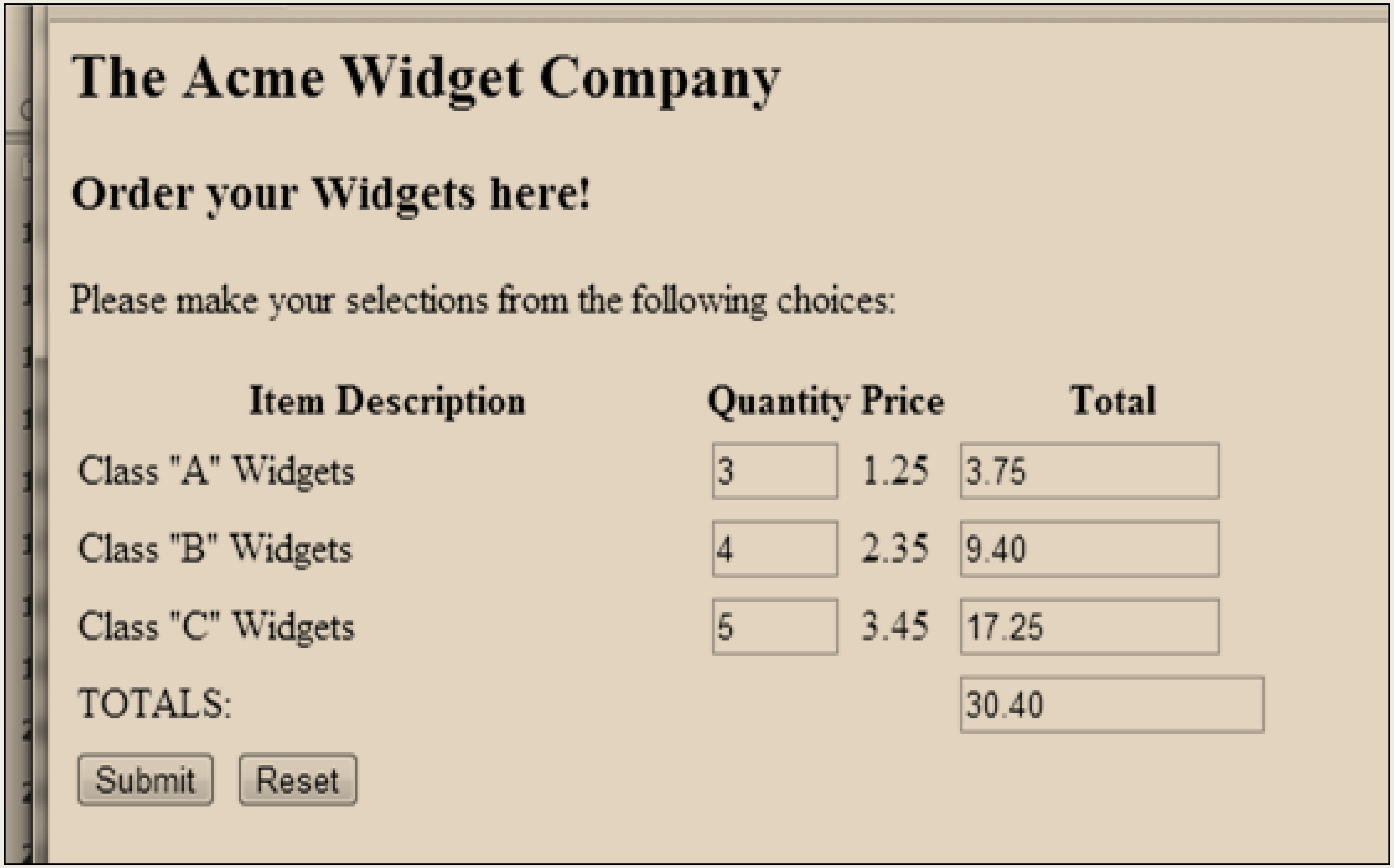
```
// ITEM B: check if user has entered a quantity in the input box
// if they have assign it to an internal variable

  if (document.getElementById("qtyB").value > "")
     {
        QtyB = document.getElementById("qtyB").value;
     }

 // calculate the subtotal for Item B & display it

   TotB = eval(QtyB) * eval(PrcB);
   TotB = TotB.toFixed(2);
    document.getElementById("totalB").value = TotB;

 // ITEM C: check if user has entered a quantity in the input box
// if they have assign it to an internal variable
```

JavaScript ….4



```
 // ITEM C: check if user has entered a quantity in the input box
// if they have assign it to an internal variable

  if (document.getElementById("qtyC").value > "")
     {
        QtyC = document.getElementById("qtyC").value;
     }

 // calculate the subtotal for Item C & display it

    TotC = eval(QtyC) * eval(PrcC);
    TotC = TotC.toFixed(2);
    document.getElementById("totalC").value = TotC;

// Keep a Running Total & display it

  Totamt = eval(TotA) + eval(TotB) + eval(TotC);
  Totamt = Totamt.toFixed(2);
  document.getElementById("GrandTotal").value = Totamt;

}


</script>
```

Order Form with Calculations



The Acme Widget Company

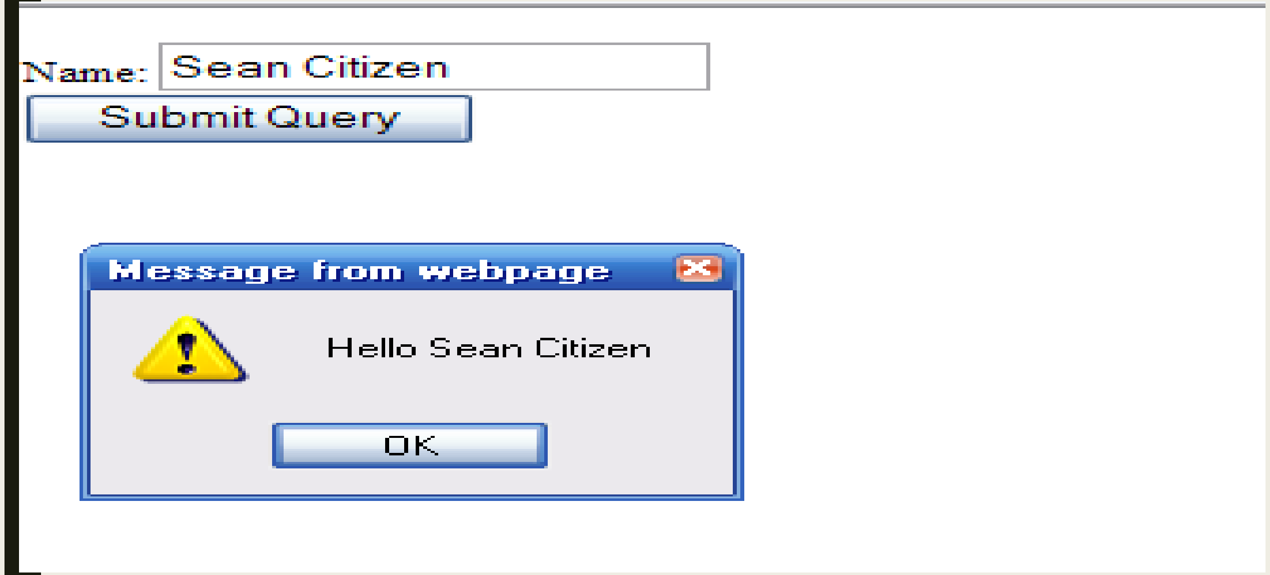
Order your Widgets here!

Please make your selections from the following choices:

Item Description	Quantity	Price	Total
Class "A" Widgets	3	1.25	3.75
Class "B" Widgets	4	2.35	9.40
Class "C" Widgets	5	3.45	17.25
TOTALS:			30.40

Form Validation

- Common for JavaScript to validate form data before sent to server
- Reduces work that server has to do.
- Common validations
  - *Required text fields must* ***not be empty***
    - Name, address
  - *Numeric fields should* ***not contain non-numeric data***
    - Telephone numbers
  - *Must have* ***valid Email addresses***
    - Includes “@” sign, ends in “.<domain>”


Name: Sean Citizen

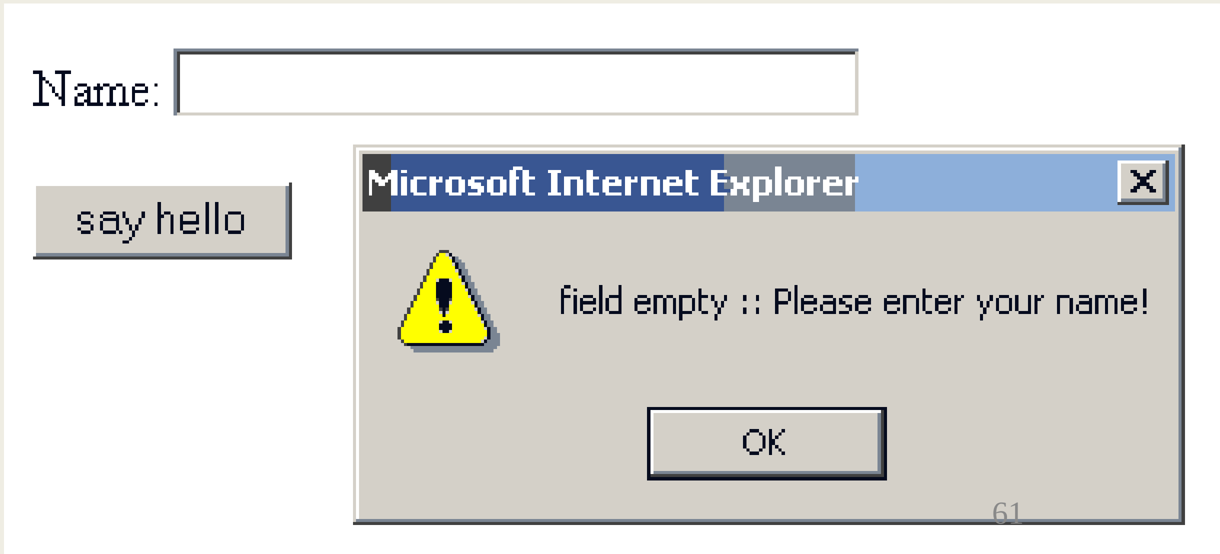
Submit Query

Message from webpage

Hello Sean Citizen

OK

Different message if form field **empty/non-empty**


Name:

say hello

Microsoft Internet Explorer

field empty :: Please enter your name!

OK

Event handler –JavaScript function

```
<body>
<!-- ************* start of form **************** --
>
<form name="form1" onSubmit="sayHello();" >
   <p>
   Name:
      <input type="text" name="name‟ id = “username”
>
   </p>
   <p>
   <input type="submit" value="say hello" >
   </p>
</form>
<!-- ************* end of form **************** -->
</body>
```

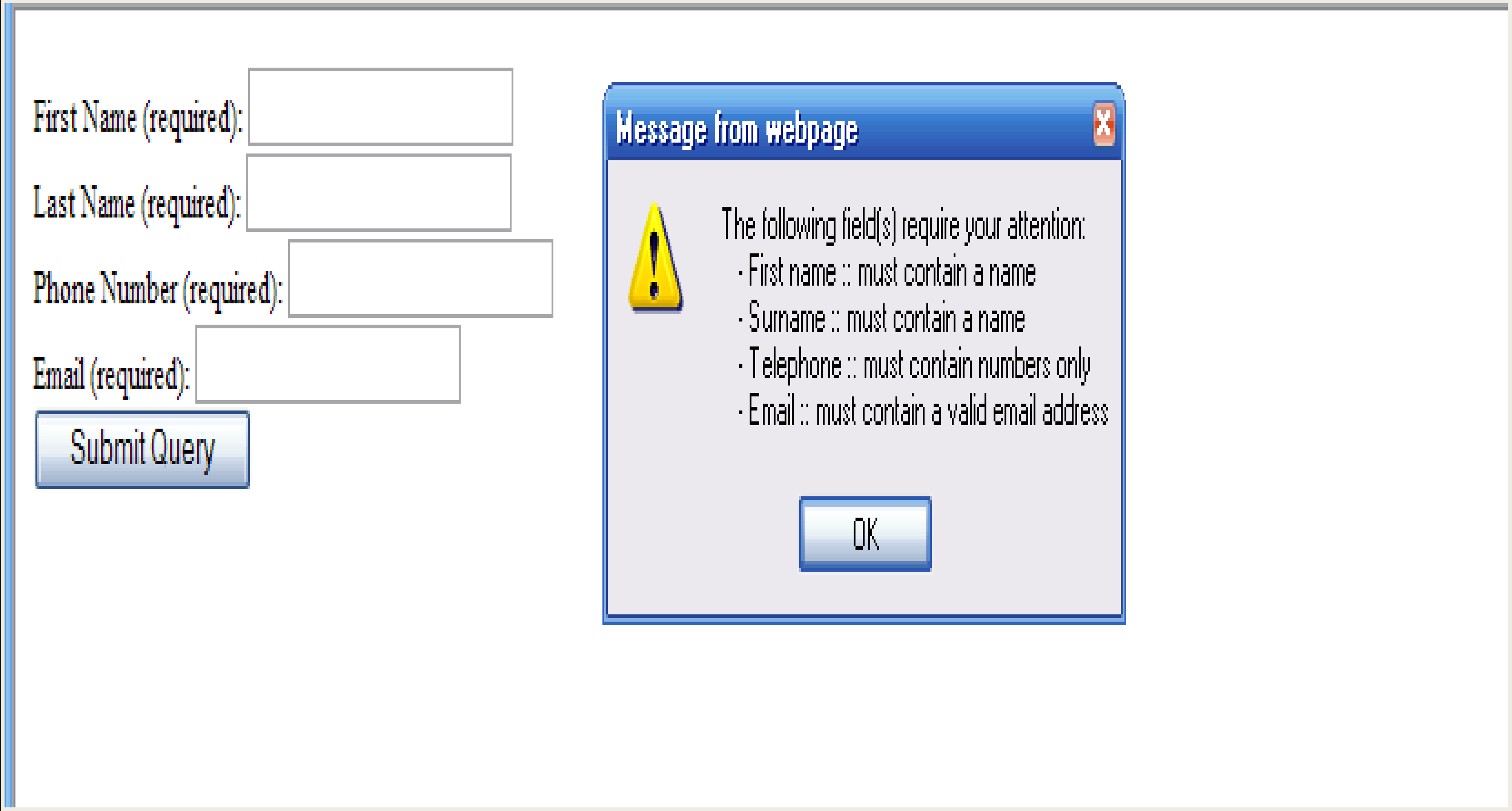
```
<html> <head>
<script type="text/javascript">
function sayHello()
{
    // get username entered on form
    var username =
             document.getElementById("username").value;
    // message for empty/non-empty form value
    if( username == "")
    {
     alert( "field empty :: Please enter your name!" );
    }
    else
    {
     alert( "Hello " + username );
    }
} // function
</script></head>          ... HTML body follows ...
```

JavaScript function defined in document <head>

- Refer to **username** textbox of the form in script as:
  - *document.getElementById(“username”)*
  - *Have to specify the “value” property to retrieve the actual text typed*
  - *document.getElementById(“username”).value*

- The value input by the user is put into a variable called username
- Then check to see if it is blank or not
- If not blank, display an Alert message including text entered in **username**: **alert(“Hello ” + username);**
- If blank, output an error message

Detailed error message about each bad form field


First Name (required):

Last Name (required):

Phone Number (required):

Email (required):

Message from webpage

The following field(s) require your attention:
- First name :: must contain a name
- Surname :: must contain a name
- Telephone :: must contain numbers only
- Email :: must contain a valid email address

```
<html>
<head>
```

Load useful function from external file

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

---

function to test all form data and return true/false

```
<script type="text/javascript">
function validateForm()
{
```

... See listing in External File overleaf!...

```
}

</script>
</head>
```

... Listing in External File overleaf!...

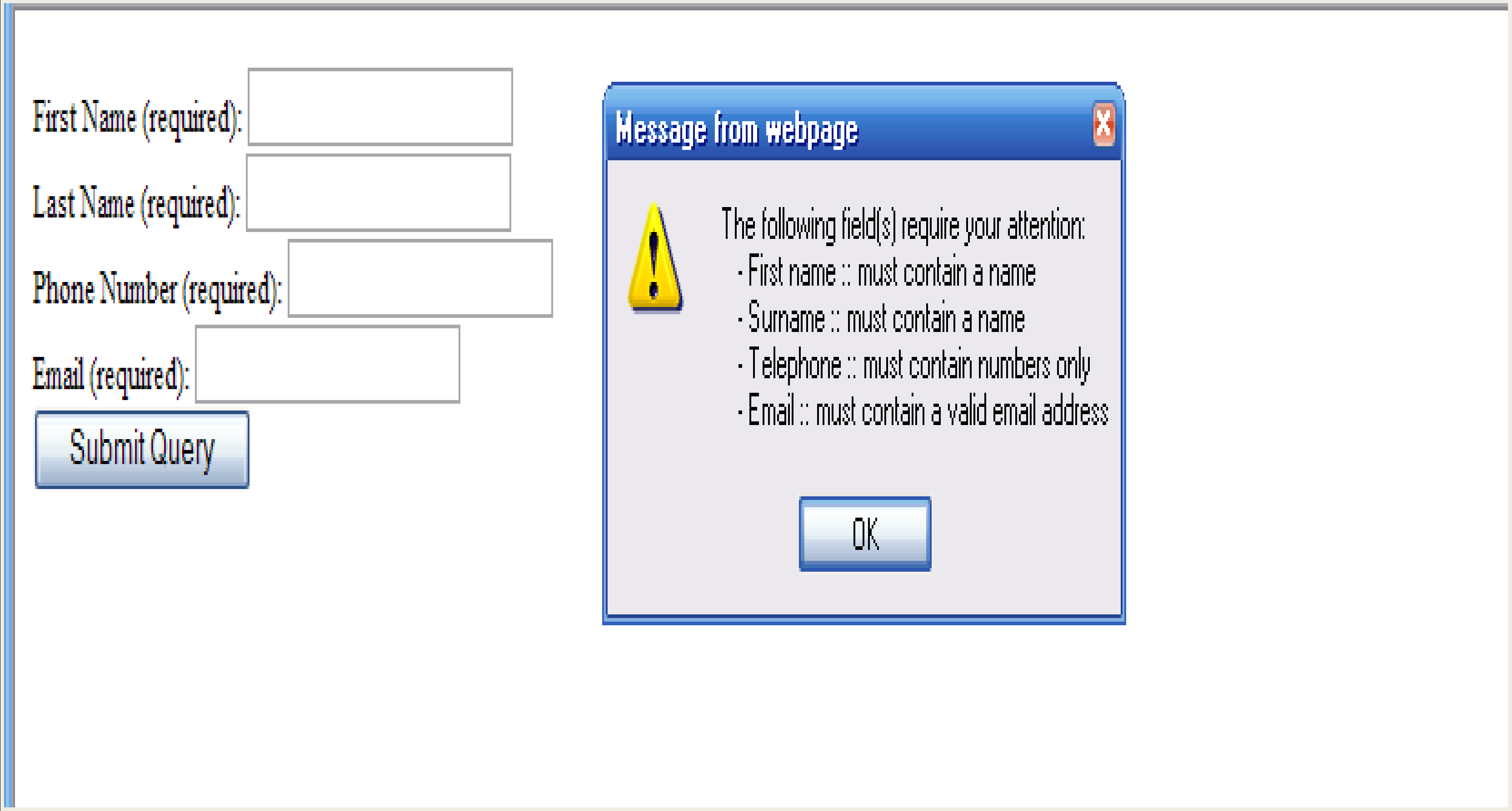
```
<script type="text/javascript">
function validateForm()
{
    // initialise NO ERRORS – GLOBAL VARIABLES – no VAR
    errorMessage = "The following field(s) require your attention:";
    hasErrors = false;
     newline = "\n   - ";

    // test firstName field
    if( isEmpty( document.getElementById(“firstName”).value ) ){
        errorMessage += newline + "First name :: must contain a name";
        hasErrors = true;
    }
    // test surname field
    if( isEmpty( document.getElementById(“surname”).value ) ){
        errorMessage += newline + "Surname :: must contain a name";
        hasErrors = true;           ..............................ETC.
    }
```

```
<form
action="page2.html“
method=“Post“      onSubmit="return validateForm();“ >
<p>
  First Name (required):
  <input type="text" name="firstName“ id = “firstName”>
  <br>
  Last Name (required):
  <input type="text" name="surname“ id = “surname”>
  <br>
  Phone number:
  <input type="text" name="telephone“ id = “telephone>
  <br>
  Email (required):
  <input type="text" name="email“ id = “email”>
  <br>
  <input type="submit">
</form>
```

Library of useful form functions

- “form_functions.js”
- A separate JavaScript text file – contains common data validation functions
- Using these will make form validation simpler
- Write <script> element in <head> of each (X)HTML page containing a form that reads in this external “.js” text file called “form_functions.js”

Detailed error message about each bad form field


First Name (required):

Last Name (required):

Phone Number (required):

Email (required):

Submit Query

Message from webpage

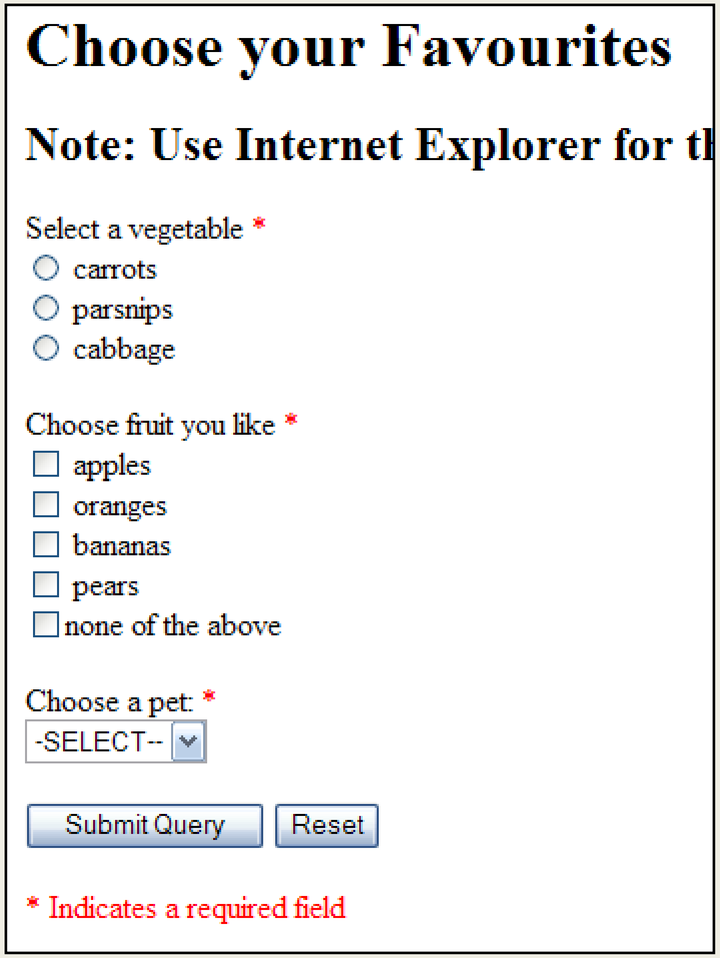
The following field(s) require your attention:
- First name :: must contain a name
- Surname :: must contain a name
- Telephone :: must contain numbers only
- Email :: must contain a valid email address

OK

GOOD NEWS IS…

html5

Radio buttons


Choose your Favourites

Note: Use Internet Explorer for th

Select a vegetable *

- ◯ carrots
- ◯ parsnips
- ◯ cabbage

Choose fruit you like *

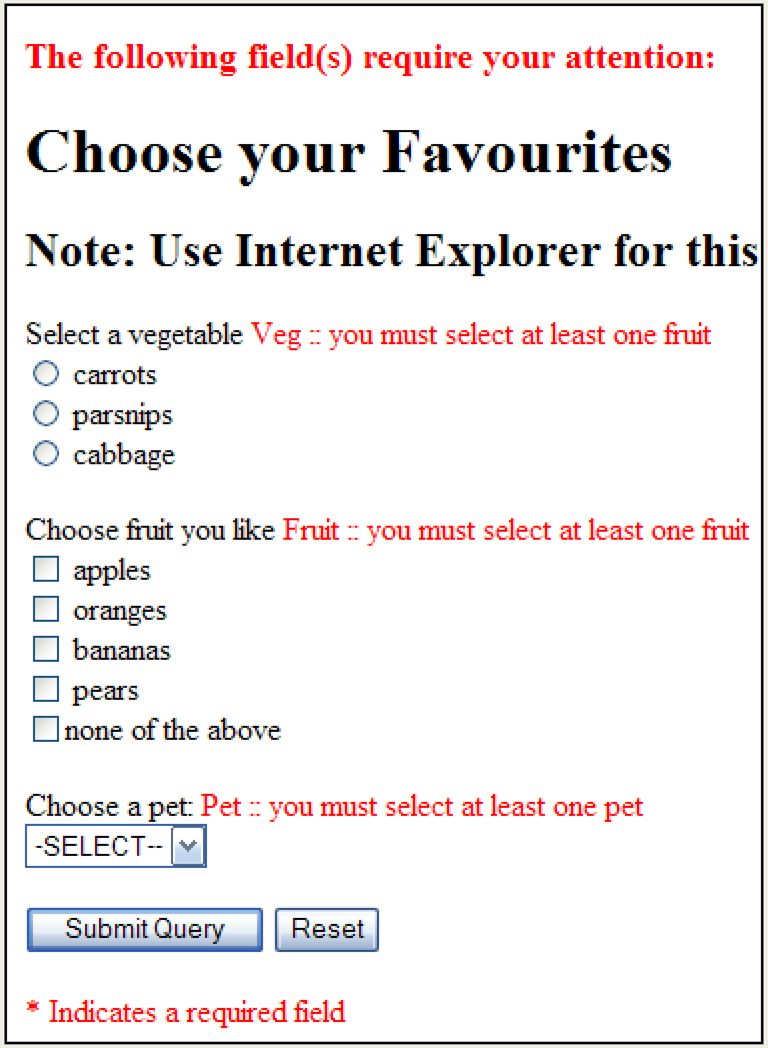
- ☐ apples
- ☐ oranges
- ☐ bananas
- ☐ pears
- ☐ none of the above

Choose a pet: *

-SELECT--

Submit Query | Reset

* Indicates a required field


The following field(s) require your attention:

Choose your Favourites

Note: Use Internet Explorer for this

Select a vegetable Veg :: you must select at least one fruit

- ◯ carrots
- ◯ parsnips
- ◯ cabbage

Choose fruit you like Fruit :: you must select at least one fruit

- ☐ apples
- ☐ oranges
- ☐ bananas
- ☐ pears
- ☐ none of the above

Choose a pet: Pet :: you must select at least one pet

-SELECT--

Submit Query | Reset

* Indicates a required field

SEE CODE

Check Dropdown Menu

```
if (document.getElementById("dropdown").selectedIndex == 0)
    {
    document.getElementById("err3").innerHTML
              = "Pet :: you must select at least one pet";
    hasErrors = true;
    }
```

•Dropdown menus are shown as a <**select**> list
•This list has a **selectedIndex** property which contains the index
•Number of the currently selected <option> list.
•Check to see, if **selectedIndex == 0** i.e. not selected
•Html name of dropdown menu = “dropdown”

Print error messages

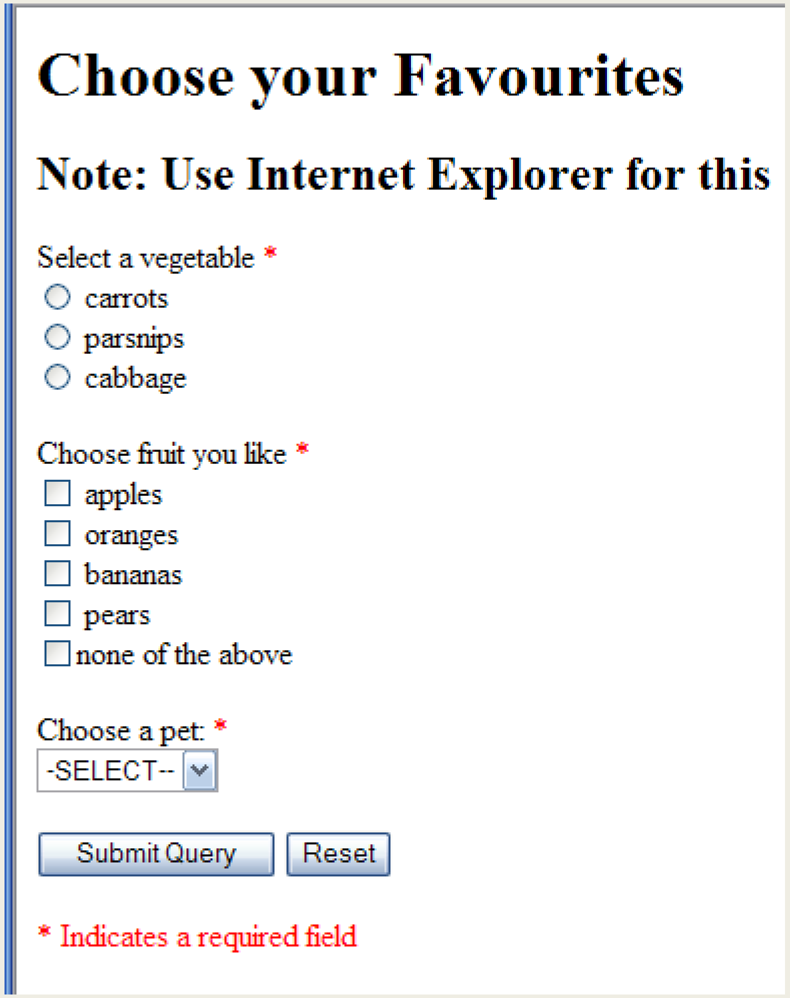
<div id = "msg"></div>
<div> blank originally
Display error message in <div> , if form invalid.

```
// do appropriate action, if errors occur – print message
    if( hasErrors ){
        document.getElementById("msg").innerHTML= "<h3>" +
    errorMessage +  "</h3>";
        return false;
         }
    else
         {
        return true;
         }
```

Onreset Event Handler

- Use the Onreset event handler to call function – **resetForm()**
- **resetForm()** removes error messages from the form and restores the form to its original format

```
<form
    name="form1"
    method="post"
    action="page2.html"
    onSubmit="return validateForm();"
    onReset ="return resetForm();"
    >
```


Choose your Favourites

Note: Use Internet Explorer for this

Select a vegetable *

- ○ carrots
- ○ parsnips
- ○ cabbage

Choose fruit you like *

- ☐ apples
- ☐ oranges
- ☐ bananas
- ☐ pears
- ☐ none of the above

Choose a pet: *

* Indicates a required field

validateForm 2 uses

- -getElementbyId
- - innerHTML
- - error messages show in <span>

Checks:

- Radio buttons
- Checkboxes
- Dropdown menu

Next weeks lab

- Forms
- Create separate file for javascript called functions.js
- In the script tag in html
- <head>
- <script src="functions.js“   type="text/javascript"></script>
- <head>

LETS LOOK AT SOME EXAMPLES OF CODE