

Client-side Processing

1

Client side processing

- Examples:
 1. HTML
 2. Plug-ins
 3. Scripts (e.g. JavaScript, VBScript, etc)
 4. Applet
 5. Cookies

- Other types of client-side processing
 1. Cascading style sheets (CSS)
 2. Dynamic HTML (DHTML)

2

Thin vs. Fat Clients

- “*where the processing takes place*”
- Early browsers were thin clients – primary function was to display web documents containing text and simple graphics
- Today’s browsers are not thin, they provide a great deal of functionality and processing on the client side
 - Scripting, active object support, email, web page authoring, audio, streaming media, Instant messaging

3

1. HTML

```
<html>
  <TITLE> HTML HST Calculator </TITLE>

  <BODY bgcolor="#FFFF80">
    <CENTER>
      <H3>HST Calculator </H3>
      To compute the amount of HST you need to pay
      take the price of the item and multiply it by 0.15
    </BODY>
</html>
```

4

Client-side interactivity

- HTML is a presentation language – *not* a programming language
- controls the appearance of the information on the client's screen but does not support processing or manipulating information
- early web browsers were limited to displaying static pages
- But ... many business applications require interactivity

5

Client-side interactivity (cont.)

- Forms were added to HTML to provide data entry
 - radio buttons, check boxed, drop-down selection lists
- Still, no processing on the client side
- Solution, embed the necessary program logic in the HTML stream and let the browser execute the instructions on the client side

6

2. Plug-ins

- A plug-in is a software routine that extends the capability of a larger application
- Allows a browser to process nonstandard, often proprietary animation, video, and audio files embedded in HTML documents

7

3. Scripts

- a Script is a set of macro-like instructions
 - JavaScript, Jscript, VBScript
- Scripts are normally embedded in an HTML document as a text string between a set of <script> tags or reference a script file between the tags
 - can control the objects, content, and interactions within the browser
 - when first introduced one of its major purposes was to validate the completeness and accuracy of data input to a browser-based form
- See <http://www.w3.org/TR/REChtml40/interact/scripts.html> for details

8

3. Scripts (continued)

- Various script functions:
 - Client-side form validation
 - Client-side calculations
 - Client-side lookup databases
 - Client-side image maps
 - Providing interactive feedback
 - Personalizing a document before it is displayed
 - Manipulating Java applets or ActiveX objects within a page

Source: Turban et al., Appendix C, Table C.1

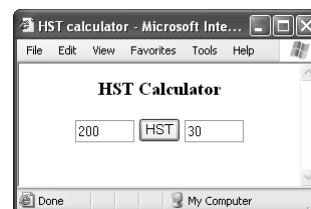
9

Forms and JavaScript

```
<HTML>
<TITLE> HST calculator </TITLE>

<SCRIPT LANGUAGE="JavaScript">
function HST(theForm)
{
    var price=parseInt(theForm.price.value);
    var hst=price*.15;
    theForm.result.value=hst;
}
</SCRIPT>

<BODY>
<CENTER>
<H3>HST Calculator </H3>
<FORM>
<input type="text" name="price" size="6" value="0">
<input type="button" value="HST" onClick="HST(this.form)">
<input type="text" name="result" size="6" value="">
</FORM>
</BODY>
</HTML>
```



<http://www.cs.dal.ca/~tt/ECMM6010/hst.htm>

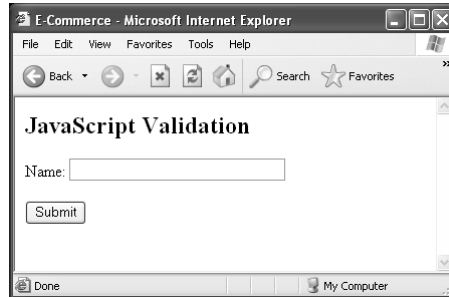
10

JavaScript example: form validation

```
<HTML>
<HEAD>
  <TITLE>E-Commerce</TITLE>
</HEAD>

<SCRIPT LANGUAGE = "JavaScript">
  function valcheck()
  {
    if(document.forms[0].elements[0].value == "")
      {alert("Please enter a name"); }
  }
</SCRIPT>

<BODY BGCOLOR = "#FFFFFF">
  <H2>JavaScript Validation</H2>
  <FORM METHOD = POST>
    <P> Name: <INPUT TYPE = "text" NAME = "T1" SIZE = "32">
    <P> <INPUT TYPE="button" NAME="B1" VALUE="Submit" onClick="valcheck()">
  </FORM>
</BODY>
</HTML>
```



<http://www.cs.dal.ca/~tt/ECMM6010/validate.htm>

11

JavaScript versions

- Originally introduced by Netscape
- Different versions of JavaScript (Jscript)
- ECMA standard ("ECMA International is an industry association founded in 1961 and dedicated to the standardization of Information and Communication Technology (ICT) Systems" <http://www.ecma-international.org>)
- Challenge of programming for different version and support of JavaScript (→ use object and method detection)

12

VBScript

```
<HTML>
<TITLE> HST calculator (VBScript) </TITLE>

<SCRIPT LANGUAGE="VBScript">

  Sub HSTButton_OnClick()
    dim price, hst
    price=CInt( Document.Forms( 0 ).price.Value)
    hst = price * .15
    Document.forms( 0 ).result.Value = hst
  End Sub

</SCRIPT>

<BODY>
<CENTER>
<H3>HST Calculator </H3>
<FORM>
  <input type="text" name="price" size="6" value="0">
  <input type="button" name="HSTButton" value="HST">
  <input type="text" name="result" size="6" value="">
</FORM>
</BODY>
</HTML>
```

13

4. Applets

- scripts are not full-featured programming languages
- to add more interactivity, applets can be downloaded to a client
- applets are small programs executed from within another program such as a browser
- an applet's file name is inserted into the body of an HTML document between <applet> tags
- an applet is stored in a web server file and downloaded to the client upon request
- most applets are written in Java
- CANNOT write to client's machine

14

Applet (cont.)

```
<html>
<TITLE> HTML and forms sample </TITLE>

<BODY bgcolor="#FFFF80">
<CENTER>
<H3>HST Calculator </H3>

To compute the amount of HST you need to pay, enter the price of the item and
click the HST Button.<br>
The information will be sent back to the server and your
GST computed

<applet code="com.mindprod.canadiantax.CanadianTaxCalculator.class"
archive="canadiantax.jar" width="600" height="400" vspace="10" hspace="10"
alt="Sorry, you need Java to run this Applet">
</applet>

</BODY>
</html>
```

[Another example with mouse interaction](#)

15

Cascading Style Sheets

- Allows you to specify the style of your web page elements separately from the structure of your document
 - “separation of presentation and content”
- Benefit of greater manageability
- Inline, Header, or External (examples)

16

Cascading style sheet - inline

```
<html>
<TITLE> CSS Inline Example </TITLE>

<BODY bgcolor="#FFFF80">
<CENTER>
<H3 STYLE = "color: #0000FF">HST Calculator </H3>

To compute the amount of HST you need to pay, take the price of the item and
multiply it by 0.15

<H3> End of Instructions </H3>

</BODY>
</html>
```

17

Cascading style sheet - header

```
<html>
<HEAD>
<TITLE> CSS Header Example </TITLE>

<STYLE TYPE = "text/css">

  H3 {color: #0000FF}
  A:hover {background-color: #CCFFCC}
  .red {color: red}

</STYLE>
</HEAD>

<BODY bgcolor="#FFFF80">
<CENTER>
<H3>HST Calculator </H3>

To compute the amount of HST you need to pay, take the price of the item and
multiply it by 0.15 or go to the <a href=javascript_example.html>javascript_example.html</A>

<H3> End of Instructions </H3>

<H3 CLASS = "red">END OF WEB PAGE </H3>

</BODY>
</html>
```

18

Cascading style sheet - external

```
<html>
<HEAD>
<TITLE> CSS External Example </TITLE>
<LINK REL = "stylesheet" TYPE = "text/css" HREF = "styles_example.css">
</HEAD>

<BODY bgcolor="#FFFF80">
<CENTER>
<H3>HST Calculator </H3>

To compute the amount of HST you need to pay, take the price of the item and
multiply it by 0.15 or go to the <a href=javascript_example.html>javascript_example.html</A>

<H3> End of Instructions </H3>

<H3 CLASS = "red">END OF WEB PAGE </H3>

</BODY>
</html>
```

styles_example.css file

```
H3 {color: #0000FF}

A:hover {background-color: #CCFFCC}

.red {color: red}
```

19

Dynamic HTML

- “allow any element of a page to be changeable at any time”
- DHTML is not a language!
- DHTML describes the abstract concept of breaking up a page into manipulatable elements, and exposing those elements to a scripting language which can perform the manipulations
- Often programmed using a combination of HTML, Cascading Style Sheets, and JavaScript
- Used for:
 1. Positioning; precisely placing blocks of content on the page and, if desired, moving these blocks around.
 2. Style modifications; on-the-fly altering the aesthetics of content on the page.
 3. Event handling; how to relate user events to changes in positioning or other style modifications.

Source: WDVL: Dynamic HTML, retrieved from
<http://www.wdvl.com/Authoring/DHTML/>, September 27, 2002

20

DHTML Object Model

```
<HTML>
<TITLE> Javascript sample </TITLE>
  <SCRIPT LANGUAGE="JavaScript">

      function GST(theForm){
        var price=parseInt(theForm.price.value)
        var gst=price*.15
        theForm.result.value=gst
        document.body.style.backgroundColor = "cyan"}

  </SCRIPT>
<BODY bgcolor="#FFFF80">
  <CENTER>
    <H3>HST Calculator </H3>
    <FORM>
      <input type="text" name="price" size="6" value="0">
      <input type="button" value="GST" onClick="GST(this.form)">
      <input type="text" name="result" size="6" value="">
    </FORM>
  </BODY>
</HTML>
```

21

DHTML Event

```
<HTML>
<TITLE> DHTML sample #2 </TITLE>
  <SCRIPT LANGUAGE="JavaScript">

      function GST(theForm){
        var price=parseInt(theForm.price.value)
        var gst=price*.15
        theForm.result.value=gst
        document.body.style.backgroundColor = "cyan"
      }

      function redtext()
      {
        calctitle.style.color = "red"
      }

  </SCRIPT>
<BODY bgcolor="#FFFF80" ONMOUSEMOVE = "redtext()">
  <CENTER>
    <H3 ID= "calctitle" >HST Calculator </H3>
    <FORM>
      <input type="text" name="price" size="6" value="0">
      <input type="button" value="GST" onClick="GST(this.form)">
      <input type="text" name="result" size="6" value="">
    </FORM>
  </BODY>
</HTML>
```

22

State

- Web is inherently stateless
 - Fine for regular browsing
- Many web applications require a series of steps
 - Requires the ability to “remember” the results of previous steps
 - Example: online banking
 - enter id and password, system authenticates you, should be able to check balances, make payments, etc. You shouldn't have to keep re-identifying yourself
 - maintaining state on server
 - imagine a site with 1,000s of hits at any instance

23

Cookies

- Maintain state by storing a small text file on the client computer
 - holds information about the user and/or the state of the application
 - when the browser transmits new data or requests a new page, the cookie is returned to the server where its contents can be used to “remind” the server of the stat

File tt@amazon[1].txt:

```
session-id-time
1064822400
amazon.com/
1536
2923479040
29591135
180892160
29589790
*
session-id
103-8845915-6364609
amazon.com/
1536
2923479040
29591135
181052160
29589790
*
ubid-main
430-9692026-3608944
amazon.com/
1536
2916341376
31961269
189172160
29589790
*
```

24

Cookies with server-side databases

- Server-side database holds a set of data that reflect the session contents
- Cookie holds the user's database access key
- Example (Davis and Benamati, pg. 89-90):
 - Amazon.ca
 1. Customer follows a hyperlink to amazon.ca
 2. Server stores a set of session data (initial contents of the shopping cart) in a database and created a session cookie containing the user's database access key
 3. Session data and cookie are returned to the client as part of the page's HTML stream
 4. Client browser stores the session data and the cookie in memory
 5. Whenever the customer changes the shopping cart, the session data and cookie are transmitted back to the server

25

Personalized Web Pages

- Store a cookie which holds the individual's information
- The cookie gets uploaded to the server when the page is requested to provide the "personal" information

Cookies & Security ???

26