Computer Networks 1
(Mạng Máy Tính 1)

Lectured by: Dr. Phạm Trần Vũ

MEng. Nguyễn Cao Đạt

Lecture 10:

Application Layer (con’t)

Reference:

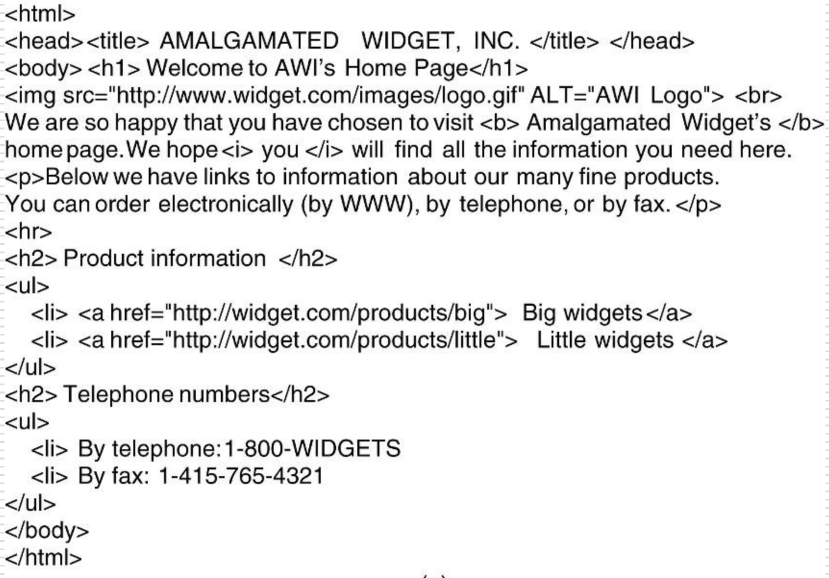
Chapter 7 - “*Computer Networks*”,
Andrew S. Tanenbaum, 4th Edition, Prentice
Hall, 2003.

Outline

- HTML- HyperText Markup Language
- Dynamic Web Documents
  - Client-Side Dynamic Web Page
  - Server-Side Dynamic Web Page

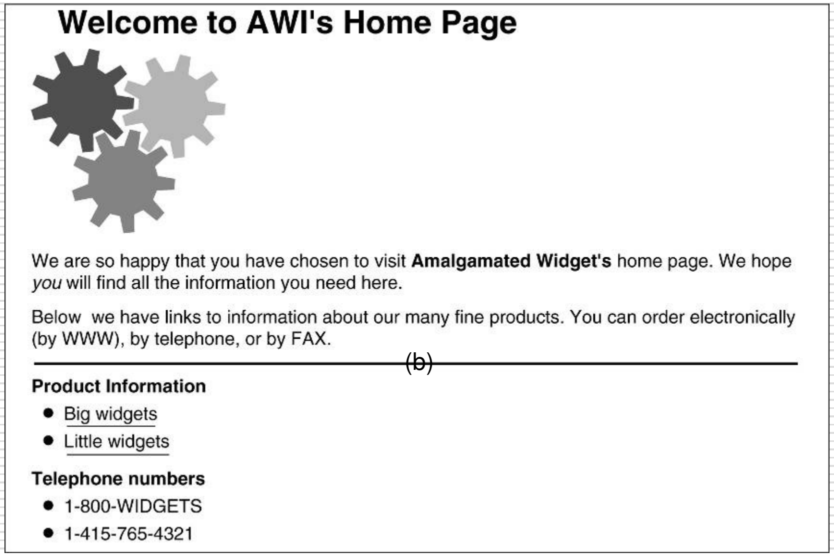
HTML – HyperText Markup Language

(a) The HTML for a sample Web page. (b) The formatted page.



```
<html>
<head><title> AMALGAMATED  WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page</h1>
<img src="http://www.widget.com/images/logo.gif" ALT="AWI Logo"> <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's </b>
home page. We hope <i> you </i> will find all the information you need here.
<p>Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information </h2>
<ul>
   <li> <a href="http://widget.com/products/big"> Big widgets </a>
   <li> <a href="http://widget.com/products/little"> Little widgets </a>
</ul>
<h2> Telephone numbers</h2>
<ul>
   <li> By telephone: 1-800-WIDGETS
   <li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```

(a)



Welcome to AWI's Home Page

We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope *you* will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

(b)

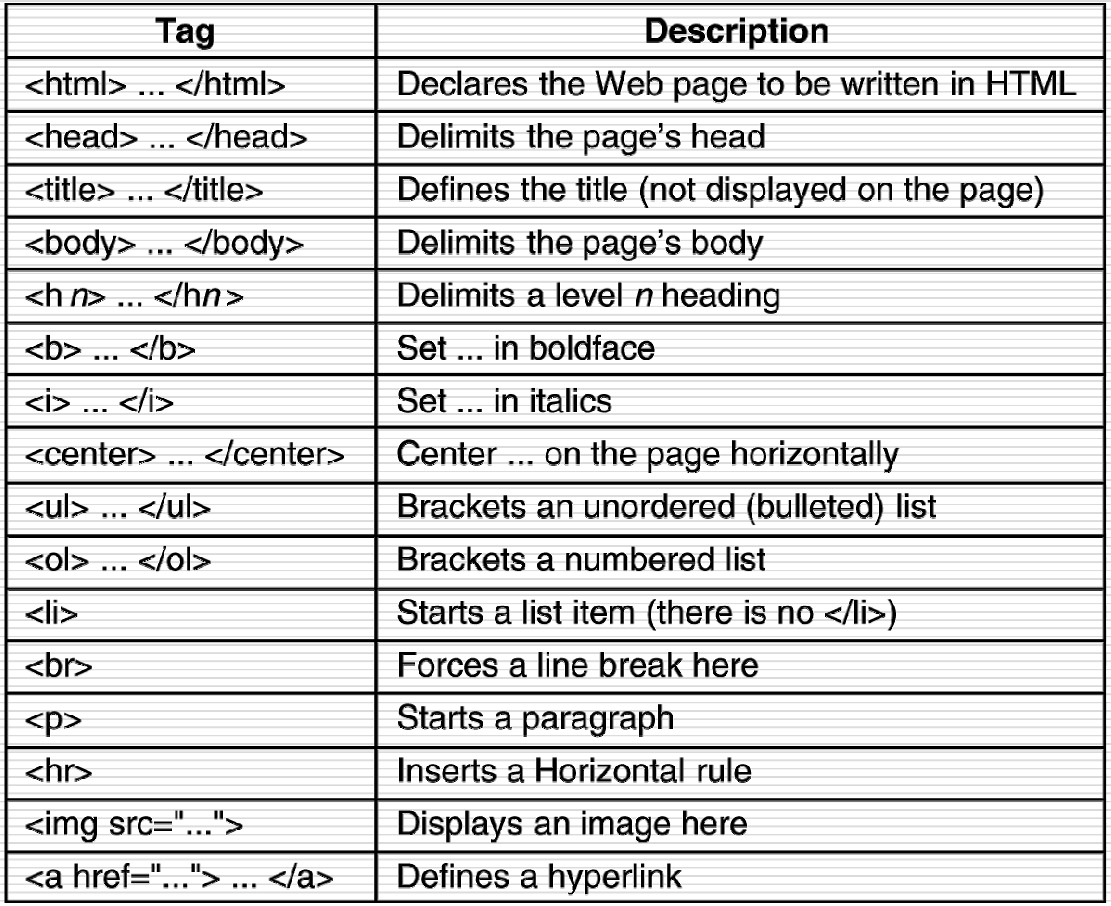
Product Information

- Big widgets
- Little widgets

Telephone numbers

- 1-800-WIDGETS
- 1-415-765-4321

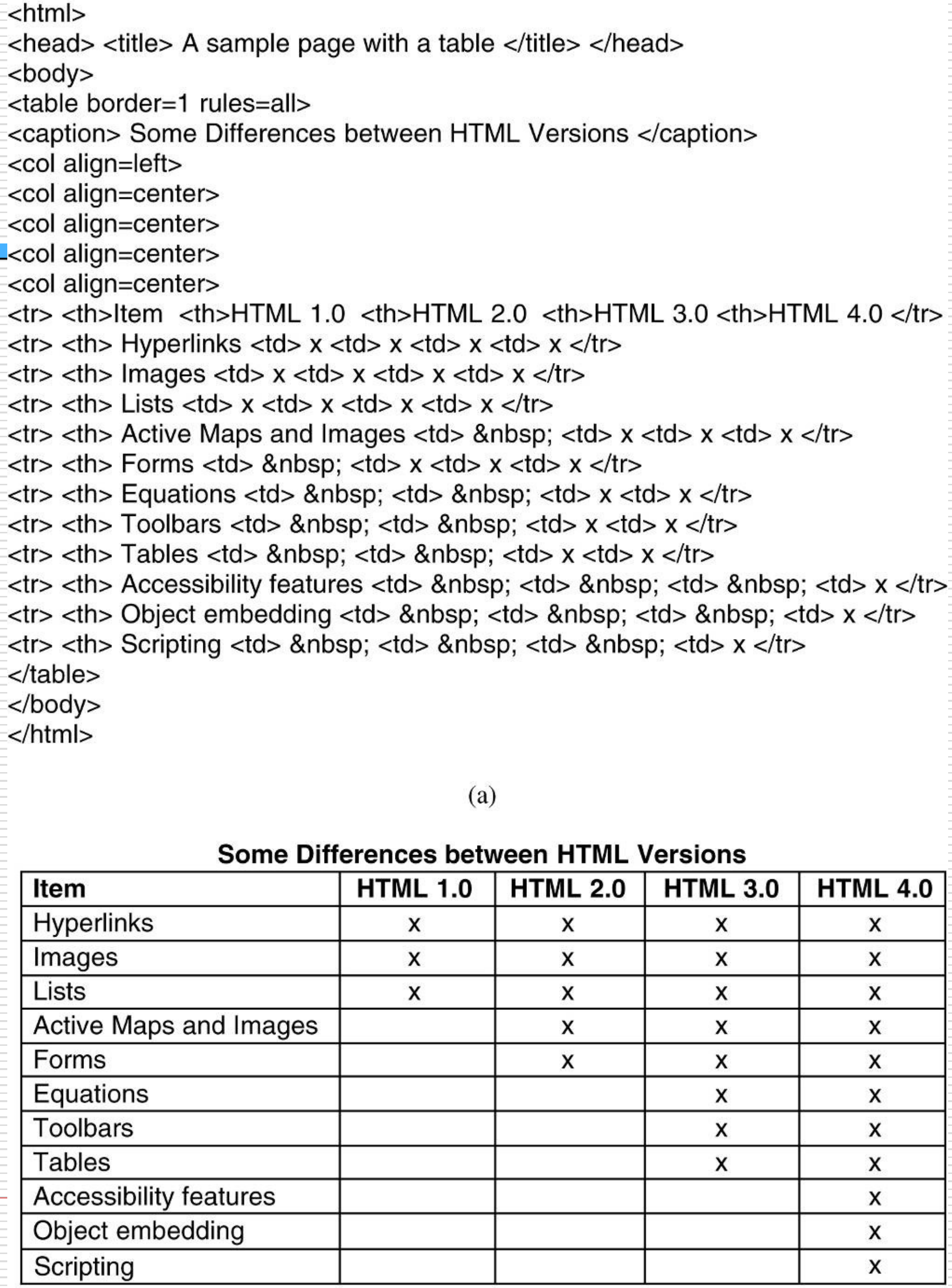
HTML Tags



Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
<b> ... </b>	Set ... in boldface
<i> ... </i>	Set ... in italics
<center> ... </center>	Center ... on the page horizontally
<ul> ... </ul>	Brackets an unordered (bulleted) list
<ol> ... </ol>	Brackets a numbered list
<li>	Starts a list item (there is no </li>)
<br>	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a Horizontal rule
<img src="...">	Displays an image here
<a href="..."> ... </a>	Defines a hyperlink

Tables

- (a) An HTML table.
- (b) A possible rendition of this table.



```
<html>
<head> <title> A sample page with a table </title> </head>
<body>
<table border=1 rules=all>
<caption> Some Differences between HTML Versions </caption>
<col align=left>
<col align=center>
<col align=center>
<col align=center>
<col align=center>
<tr> <th>Item  <th>HTML 1.0  <th>HTML 2.0  <th>HTML 3.0 <th>HTML 4.0 </tr>
<tr> <th> Hyperlinks <td> x <td> x <td> x <td> x </tr>
<tr> <th> Images <td> x <td> x <td> x <td> x </tr>
<tr> <th> Lists <td> x <td> x <td> x <td> x </tr>
<tr> <th> Active Maps and Images <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Forms <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Equations <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Toolbars <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Tables <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Accessibility features <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Object embedding <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Scripting <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
</table>
</body>
</html>
```

(a)

Some Differences between HTML Versions

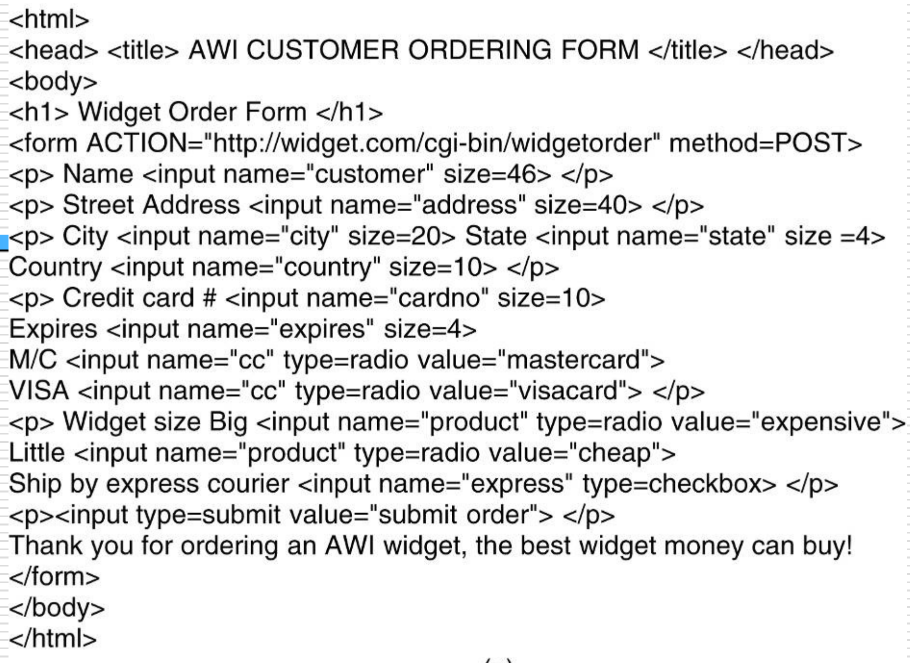
Item	**HTML 1.0**	**HTML 2.0**	**HTML 3.0**	**HTML 4.0**
Hyperlinks	x	x	x	x
Images	x	x	x	x
Lists	x	x	x	x
Active Maps and Images		x	x	x
Forms		x	x	x
Equations			x	x
Toolbars			x	x
Tables			x	x
Accessibility features				x
Object embedding				x
Scripting				x

(b)

Forms

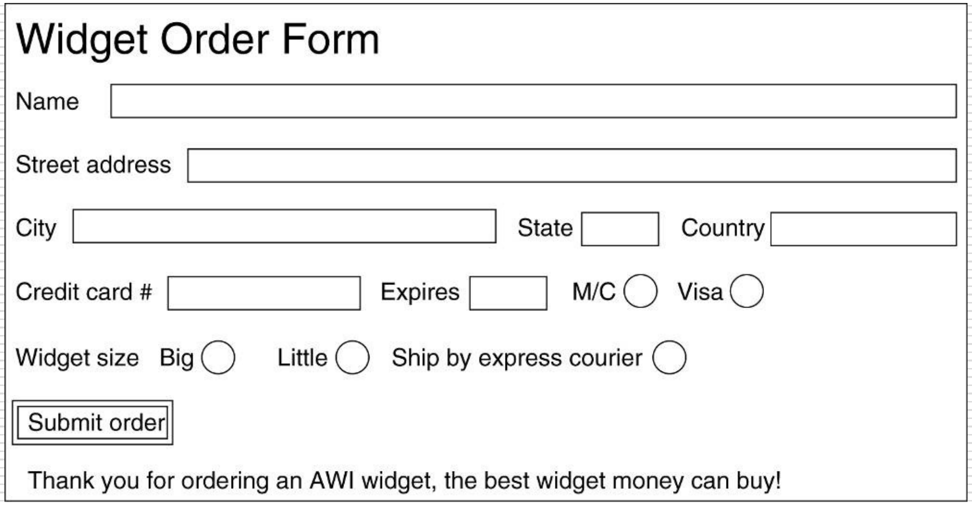
(a) The HTML for an order form.

(b) The formatted page.



```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/widgetorder" method=POST>
<p> Name <input name="customer" size=46> </p>
<p> Street Address <input name="address" size=40> </p>
<p> City <input name="city" size=20> State <input name="state" size =4>
Country <input name="country" size=10> </p>
<p> Credit card # <input name="cardno" size=10>
Expires <input name="expires" size=4>
M/C <input name="cc" type=radio value="mastercard">
VISA <input name="cc" type=radio value="visacard"> </p>
<p> Widget size Big <input name="product" type=radio value="expensive">
Little <input name="product" type=radio value="cheap">
Ship by express courier <input name="express" type=checkbox> </p>
<p><input type=submit value="submit order"> </p>
Thank you for ordering an AWI widget, the best widget money can buy!
</form>
</body>
</html>
```

(a)



Widget Order Form

Name

Street address

City State Country

Credit card # Expires M/C ○ Visa ○

Widget size Big ○ Little ○ Ship by express courier ○

Submit order

Thank you for ordering an AWI widget, the best widget money can buy!

(b)

Forms (2)

- Basic structure

  <FORM ACTION=*“file”* METHOD={GET|POST}>

  [<INPUT TYPE=“” NAME=“” VALUE="">]+

  </FORM>

- Input types
  - TEXT

  <INPUT [TYPE=TEXT] NAME="*text-id*" [SIZE=*nn*] [MAXLENGTH=*nn*] [VALUE="*default text*"]>

Forms (3)

- Input types (cont.)
  - SUBMIT

    <INPUT TYPE=SUBMIT [NAME="*button-id*"]
    [VALUE="*text*"]>
  - RESET

    <INPUT TYPE=RESET [VALUE=*“text"*]>
  - BUTTON

    <INPUT TYPE=BUTTON [NAME="*button-id*“]
    [VALUE="*text*"]>

Forms (4)

- Input types (cont.)
  - RADIO

    <INPUT TYPE=RADIO NAME="*radio-set-id*" VALUE="*choice-id*" [checked]>

    Ex:

    ```
    <input type="radio" value="V1" checked name="R1">Option 1
    <br><input type="radio" value="V2" name="R1">Option 2
    <br><input type="radio" value="V3" name="R1">Option 3
    ```

  - CHECKBOX

    <INPUT TYPE=CHECKBOX NAME="*id*“ VALUE="*choice-id*" [CHECKED]>

    Ex:

    ```
    <input type="checkbox" name="C1“ value="ON">Check 1 <br>
    <input type="checkbox" name="C2" value="ON" checked>Check 2
    ```

Forms (4)

- Input types (cont.)
  - HIDDEN

<INPUT TYPE=HIDDEN NAME="*id*" VALUE="*data*"**>**

- TEXTAREA

<TEXTAREA NAME="*id*" [COLS=*nn*] [ROWS=*nn*]>*default text*</TEXTAREA>

- SELECT

<SELECT NAME="*id*" [SIZE=*nn*] [MULTIPLE]>

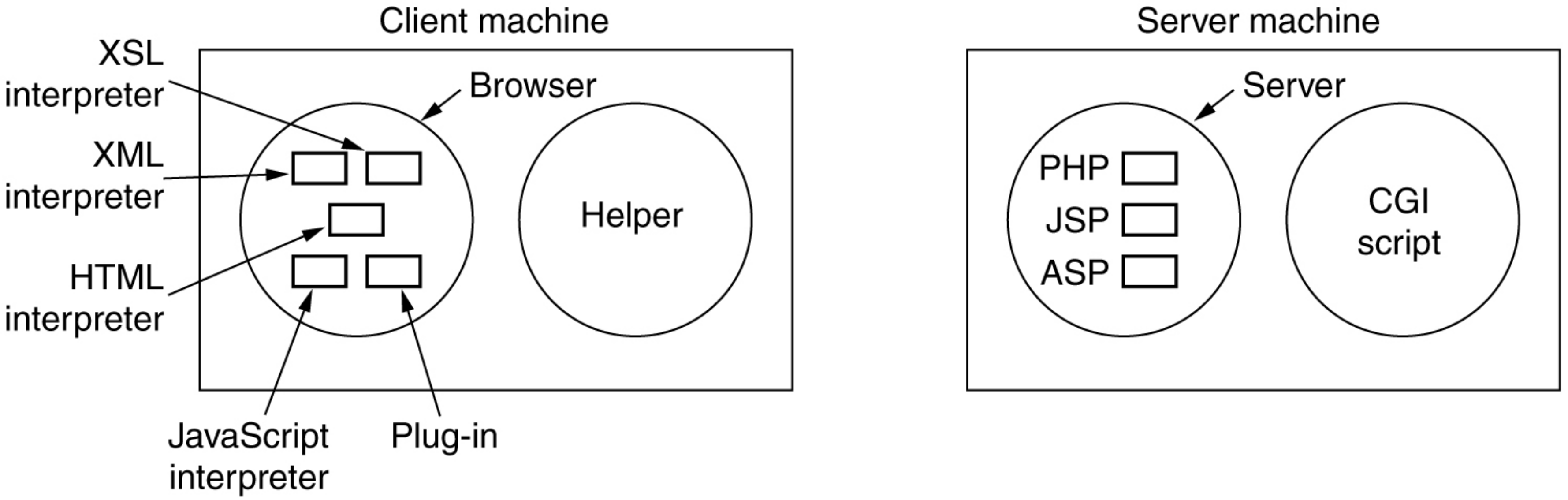
[<OPTION [VALUE="*value*"] [SELECTED]>*text*]+

</SELECT>

Dynamic Web Documents

- □ Web contents are generated dynamically on demand
- □ Dynamic Web documents are now popular in the Internet
- □ Dynamic contents can be generated on client side or/and server side

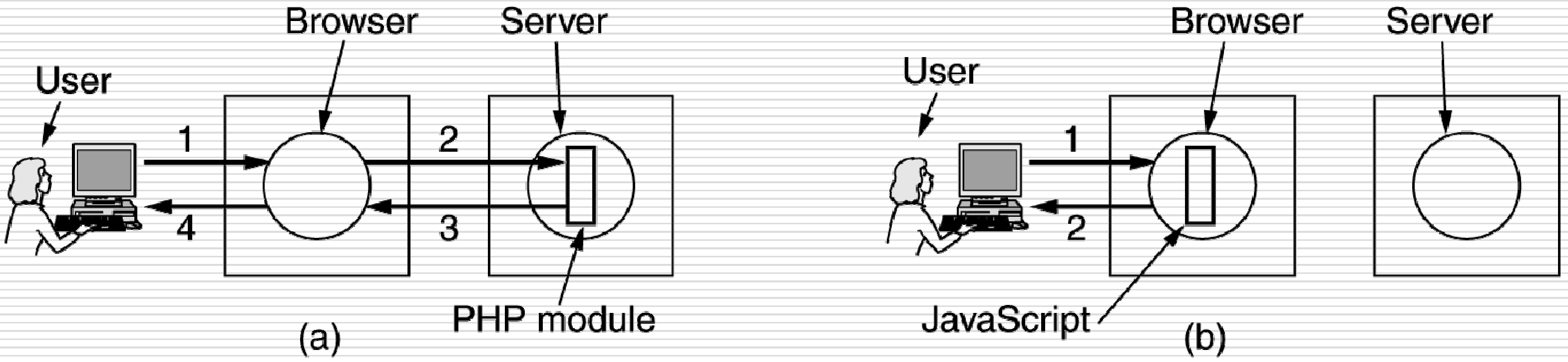
Dynamic Web Page Generation



Dynamic Web Page Generation (2)

(a) Server-side scripting with PHP.

(b) Client-side scripting with Javascript.



Client-Side Dynamic Web Page Generation with Javascript

- **Javascript code**

```
<script language=“JavaScript”>
    [var variable;]*
    function function-name([agrv]*){
        //commands;
        [return [value];]
    }
</script>
```

- **Using existed Javascript file (*.js)**

```
<script language="JavaScript" src="*.js"></script>
```

Client-Side Dynamic Web Page Generation with Javascript (2)

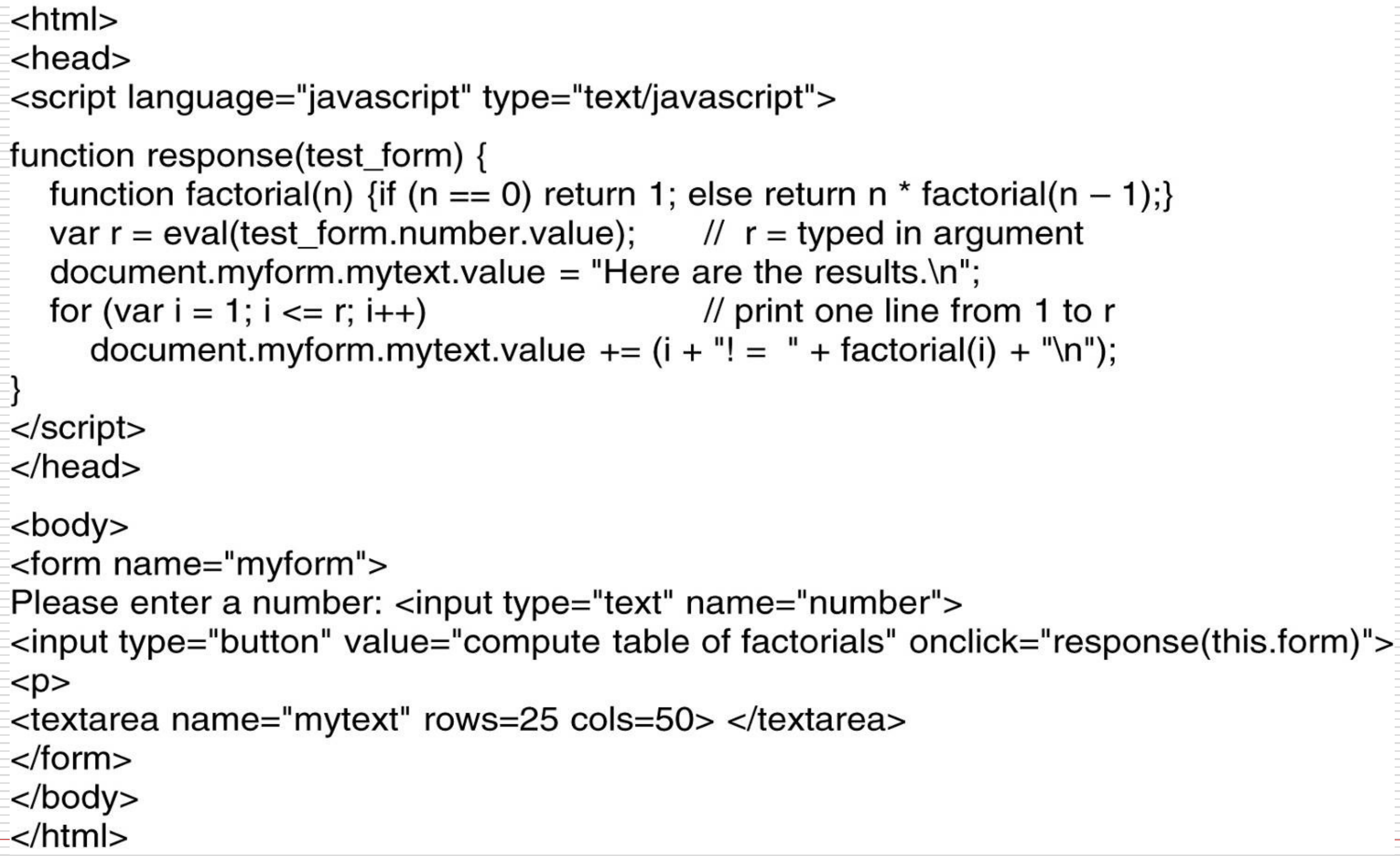
Use of JavaScript for processing a form.

```
<head>
<script language="javascript" type="text/javascript">
function response(test  form) {
   var person = test  form.name.value;
   var years = eval(test  form.age.value) + 1;
   document.open();
   document.writeln("<html>  <body>");
   document.writeln("Hello  " + person + ".<br>");
   document.writeln("Prediction:  next year you will be " + years + ".");
   document.writeln("</body> </html>");
   document.close();
}
</script>
</head>

<body>
<form>
Please enter your name: <input type="text" name="name">
<p>
Please enter your age: <input type="text" name="age">
<p>
<input type="button" value="submit" onclick="response(this.form)">
</form>
</body>
</html>
```

Client-Side Dynamic Web Page Generation with Javascript (3)

A JavaScript program for computing and printing factorials.



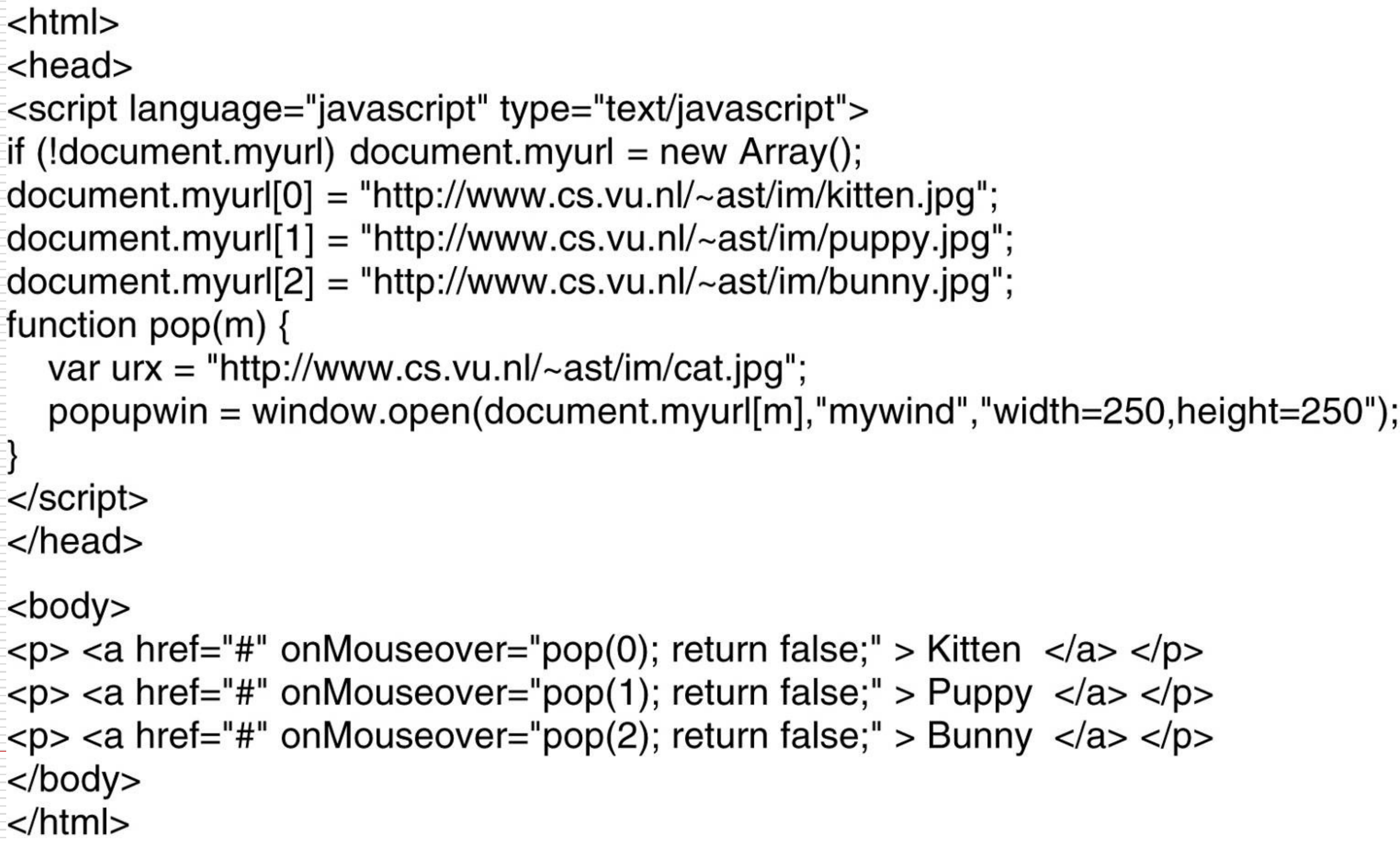
```
<html>
<head>
<script language="javascript" type="text/javascript">

function response(test_form) {
   function factorial(n) {if (n == 0) return 1; else return n * factorial(n – 1);}
   var r = eval(test_form.number.value);       //  r = typed in argument
   document.myform.mytext.value = "Here are the results.\n";
   for (var i = 1; i <= r; i++)                      // print one line from 1 to r
      document.myform.mytext.value += (i + "! =  " + factorial(i) + "\n");
}
</script>
</head>

<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute table of factorials" onclick="response(this.form)">
<p>
<textarea name="mytext" rows=25 cols=50> </textarea>
</form>
</body>
</html>
```

Client-Side Dynamic Web Page Generation with Javascript (4)

An interactive Web page that responds to mouse movement.



```
<html>
<head>
<script language="javascript" type="text/javascript">
if (!document.myurl) document.myurl = new Array();
document.myurl[0] = "http://www.cs.vu.nl/~ast/im/kitten.jpg";
document.myurl[1] = "http://www.cs.vu.nl/~ast/im/puppy.jpg";
document.myurl[2] = "http://www.cs.vu.nl/~ast/im/bunny.jpg";
function pop(m) {
   var urx = "http://www.cs.vu.nl/~ast/im/cat.jpg";
   popupwin = window.open(document.myurl[m],"mywind","width=250,height=250");
}
</script>
</head>

<body>
<p> <a href="#" onMouseover="pop(0); return false;" > Kitten  </a> </p>
<p> <a href="#" onMouseover="pop(1); return false;" > Puppy  </a> </p>
<p> <a href="#" onMouseover="pop(2); return false;" > Bunny  </a> </p>
</body>
</html>
```

Client-Side Dynamic Web Page Generation with Javascript (5)

```
function isEmail() {
if (document.forms[0].elements[1].value == '') {
alert ("\n The E-Mail field is blank. \n\n “+
        “Please enter your E-Mail address.")
document.forms[0].elements[1].focus();
return false;
}
if (document.forms[0].elements[1].value.indexOf ('@',0) == -1 ||
document.forms[0].elements[1].value.indexOf ('.',0) == -1) {
alert ("\n The E-Mail field requires a \"@\" and a \".\""+
       "be used. \n\nPlease re-enter your E-Mail address.")
document.forms[0].elements[1].select();
document.forms[0].elements[1].focus();
return false;
}
return true;
}
```

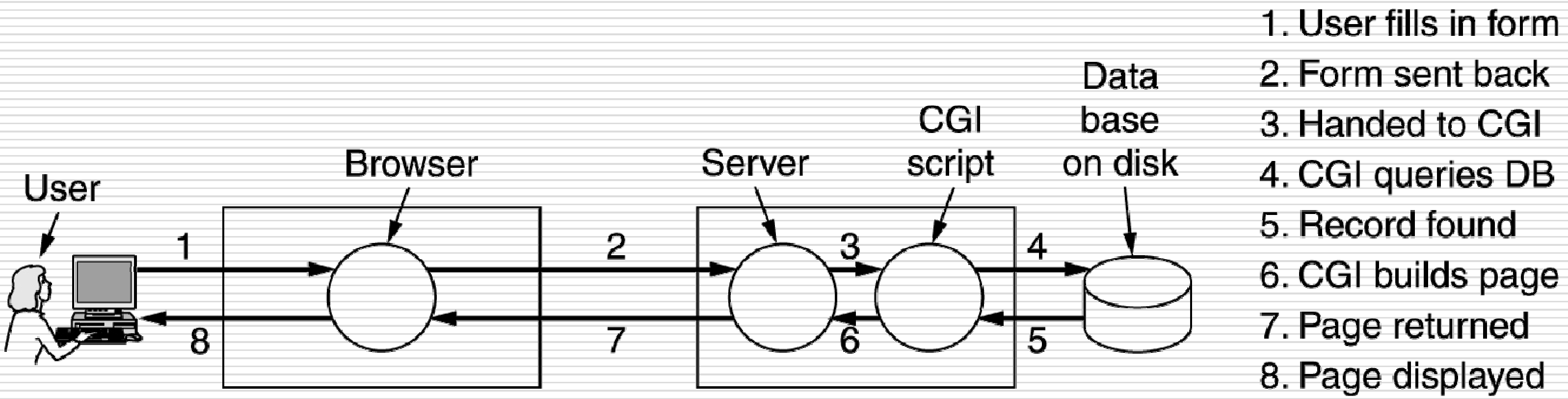
Client-Side Dynamic Web Page Generation with Java Applet

```
//file SampleApplet.java
import java.applet.*; import java.awt.*;
public class SampleApplet extends Applet {
String text = "error"; int x = 0; int y = 20;
   public void init() {
       text = getParameter("text");
       try {  x = Integer.parseInt(getParameter("x"));
          y = Integer.parseInt(getParameter("y"));
       }catch(NumberFormatException ex){ }
   }
   public void paint(Graphics g) {
       g.setFont(new Font("TimesRoman",Font.BOLD+
                 Font.ITALIC,36));
       g.drawString(text,x,y);
   }
}
```

Client-Side Dynamic Web Page Generation with Java Applet (2)

```
<HTML> <HEAD> <TITLE>Using the Applet Tag
</TITLE> </HEAD>
<BODY>
<H1>An Applet that Displays Text at a Designated
   Location</H1>
<APPLET CODE="SampleApplet.class" HEIGHT=300 WIDTH=300>
<PARAM NAME="text" VALUE="Applets are fun!">
<PARAM NAME="x" VALUE="50">
<PARAM NAME="y" VALUE="50">
Text displayed by browsers that are not Java-enabled.
</APPLET>
</BODY>
</HTML>
```

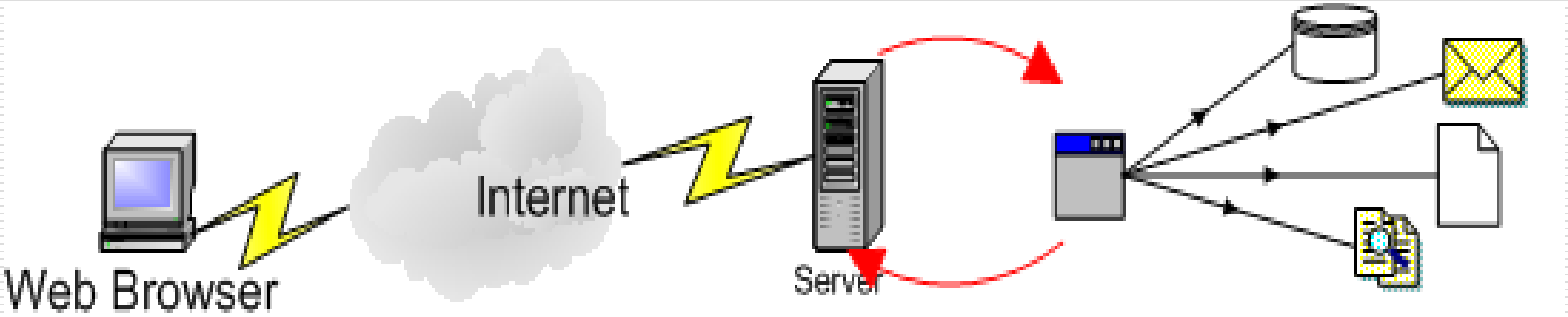
Server Side Dynamic Web Documents



1. User fills in form
2. Form sent back
3. Handed to CGI
4. CGI queries DB
5. Record found
6. CGI builds page
7. Page returned
8. Page displayed

Steps in processing the information from an HTML form.

Server Side Dynamic Web Documents with CGI using Perl



Client sends a standard HTTP request encoded with special variables and their values. These variables are commonly called "CGI variables"

The web server receives the HTTP request and using special rules, determines that the request is destined for a CGI program and not for an HTML file.

The Perl interpreter accepts the input from the server, parses the variables, applies some programming logic to the data and returns a document back to the webserver.

Perl is able to call libraries and other network functions to complete the request.

Server Side Dynamic Web Documents with CGI using Perl (2)

- Example

```
#!/perl/bin/perl
#Remember : this path will vary depending on
#where Perl is located
print "Content-type:text/html\n\n";
print "<html><head><title>HELLO!</title></head>";
print "<body>\n";
print "<h2>Hello!</h2>\n";
foreach $key (sort(keys %ENV)) {
   print "VARIABLE $key = $ENV{$key}<br>\n";
}
print "</body></html>\n";
```

Server Side Dynamic Web Documents with CGI using Perl (3)

- GET method

```
#!/perl/bin/perl
print "Content-type:text/html\n\n";
print "<html><head><title>HELLO!</title></head>";
print "<body>\n";
print "<h2>Hello!</h2>\n";
print "String = $ENV{'QUERY_STRING'}\n\n<p>";
@values = split(/&/, $ENV{'QUERY_STRING'});
foreach $i (@values) {
    ($varname, $mydata) = split(/=/, $i);
    print "The value of $varname is $mydata\n\n<p>";
}
print "</body></html>\n";
```

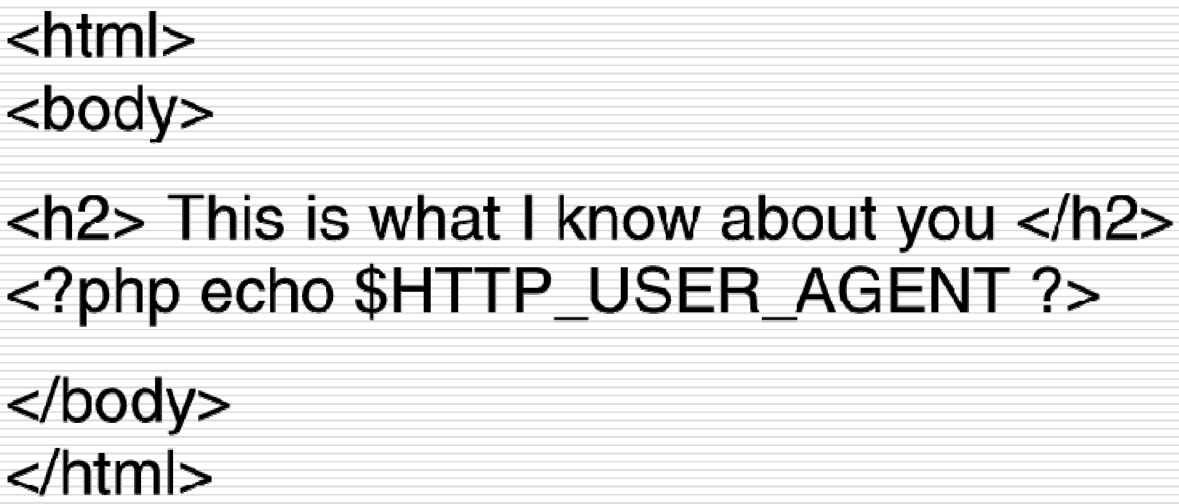
Server Side Dynamic Web Documents with CGI using Perl (4)

- POST method

```
#!/perl/bin/perl
print "Content-type:text/html\n\n";
print "<html><head><title>HELLO!</title></head>";
print "<body>\n";
print "<h2>Hello!</h2>\n";
read(STDIN, $line, $ENV{'CONTENT_LENGTH'});
@values = split(/&/, $line);
print "<h2>These were the values sent</h2>\n";
foreach $i (@values) {
    ($varname, $data) = split(/=/, $i);
    print "The value of $varname is $data\n\n<p>";
}
print "</body></html>\n";
```

Server Side Dynamic Web Documents with PHP

A sample HTML page with embedded PHP.



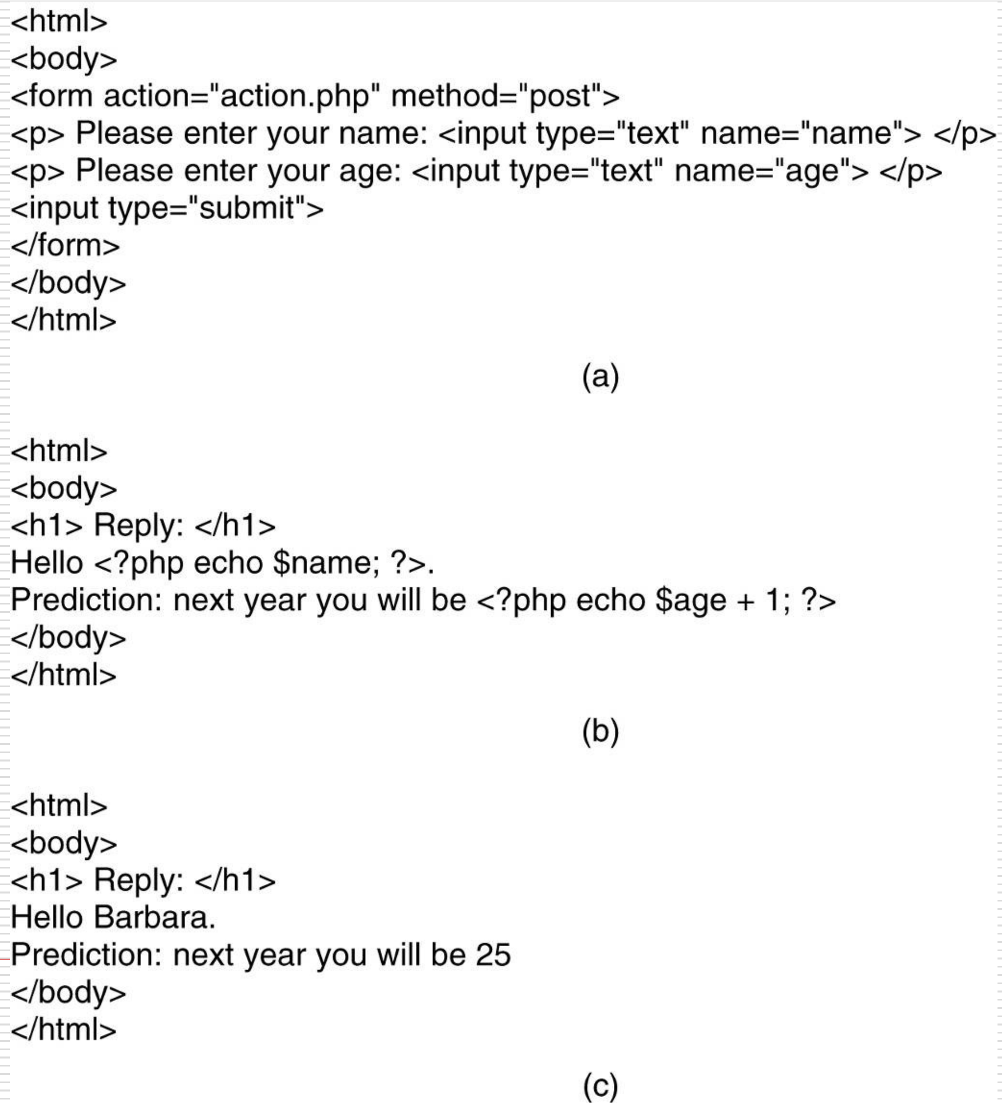
```
<html>
<body>

<h2> This is what I know about you </h2>
<?php echo $HTTP_USER_AGENT ?>

</body>
</html>
```

Server Side Dynamic Web Documents with PHP (2)

(a) A Web page containing a form. (b) A PHP script for handling the output of the form. (c) Output from the PHP script when the inputs are "Barbara" and 24 respectively.



```
<html>
<body>
<form action="action.php" method="post">
<p> Please enter your name: <input type="text" name="name"> </p>
<p> Please enter your age: <input type="text" name="age"> </p>
<input type="submit">
</form>
</body>
</html>
```

(a)

```
<html>
<body>
<h1> Reply: </h1>
Hello <?php echo $name; ?>.
Prediction: next year you will be <?php echo $age + 1; ?>
</body>
</html>
```

(b)

```
<html>
<body>
<h1> Reply: </h1>
Hello Barbara.
Prediction: next year you will be 25
</body>
</html>
```

(c)

Server Side Dynamic Web Documents with PHP (3)

- **Using loop**

```
<html><body>do..while loops<br>
<table border=1><tr><td>Quatity</td><td>Amount</td></tr>
<?php
    $soluong=5;
    $dongia=5000;
    do{
      echo "<tr><td>".$soluong."</td><td>".$soluong*$dongia.
           “</td></tr>";
        $soluong--;
    }
    while($soluong>0)
?></table></body></html>
```

Server Side Dynamic Web Documents with PHP (4)

- □ **Using Array**

```
<html><body>Use while, each and list function<br>
<?php
    $n=10;
    $Items=array($n);
    for ($i=0;$i<$n;$i++)   $Items[$i]=$i*2;
    echo "Key  Value<br>";
    while (list($k,$v)=each($Items)) {
    echo $k;
    echo "&nbsp; &nbsp;   &nbsp; &nbsp;";
    echo $v."<br>";
    }
?>
</body></html>
```

Server Side Dynamic Web Documents with PHP (5)

- **Using Array** (cont.)

```
<html><body>Sort Multidimential array<br>
<?php
    function compare($x,$y){
     if($x[0]==$y[0])  return 0;
     else if($x[0]< $y[0]) return -1;
     else  return 1;
    }
    $products=array(array("TIR","Tires", 100), array("COR","Concord", 10),
                    array("BOE","Boeing", 5000));
    usort($products,compare);
    for ($row=0;$row<3;$row++){
      for ($col=0;$col<3;$col++)
         echo "\t|\t".$products[$row][$col];
      echo "|<br>";
    }
?></body></html>
```

Server Side Dynamic Web Documents with PHP (6)

- **Session**

```
<?php
   session_start();
   $id="admin";
   $email="admin@hcmut.edu.vn";
   $fullname=“Nguyen Van A";
   session_register("id");
   session_register("email","fullname");
?>
<html><head> <title>Start session and
   register</title></head><body>
3 sessions has been registered. To view it
<a href="viewSession.php">Click
   Here</a></body></html>
```

PHP Version 4 and 5

- **Session**

```
<?php
   session_start();
   $_SESSION[‘id’]=“admin";
   $_SESSION[‘email’]="admin@hcmut.edu.vn";
   $_SESSION[‘fullname’]=“Nguyen Van A";
?>
<html><head> <title>Start session and
   register</title></head><body>
3 sessions has been registered. To view it
<a href="viewSession.php">Click
   Here</a></body></html>
```

Server Side Dynamic Web Documents with PHP (7)

- □ **viewSession.php**

```
<?php
   session_start();
?>
<html><head> <title>Get session that
   registered</title></head>
<body>
<?php
   echo "Id: $id.<br>";
   echo "Email: $email.<br>";
   echo "Fullname: $fullname.<br>";
?>
To delete session <a href="deleteSession.php"> Click
   here</a>
</body></html>
```

PHP Version 4 and 5

- **viewSession.php**

```
<?php
   session_start();
?>
<html><head> <title>Get session that
   registered</title></head>
<body>
<?php
   echo "Id:” . $_SESSION[‘id’] “.<br>";
   echo "Email:” . $_SESSION[‘email’] . “.<br>";
   echo "Fullname:” . $_SESSION[‘fullname’] . “.<br>";
?>
To delete session <a href="deleteSession.php"> Click
   here</a>
</body></html>
```

Server Side Dynamic Web Documents with PHP (8)

- deleteSession.php

```
<?php
   session_start();
?><html><head><title>delete sessions</title></head><body>
<?php
   echo "Id: $id.<br>";
   echo "Email: $email.<br>";
   echo "Fullname: $fullname.<br>";
   $result=session_is_registered("id");
   if($result==1){
   session_unregister("id");
   }
?>
To deregister all sessions or destroy session
<a href="deleteAll.php">Click here</a></body></html>
```

PHP Version 4 and 5

- deleteSession.php

```
<?php
  session_start();
?><html><head><title>delete
  sessions</title></head><body>
<?php
  echo "Id:” . $_SESSION(‘id’)
  “.<br>";
  echo "Email:” . $_SESSION(‘email’)
  . “.<br>";
  echo "Fullname:” .
  $_SESSION(‘fullname’) . “.<br>";
  if (isset($_SESSION["id“])){
       unset($_SESSION["id“]);
```

Server Side Dynamic Web Documents with PHP (9)

- deleteAll.php

```
<?php
   session_start();
   session_unset();
   session_destroy();
?>
<html><head> <title>Unregister sessions</title>
</head><body>
To Register session <a href="registerSession.php">
click me</a>
 </body></html>
```

PHP Version 4 and 5

- deleteAll.php

```
<?php
   session_start();
   $_SESSION=array();
   session_destroy();
?>
<html><head> <title>Unregister sessions</title>
</head><body>
To Register session <a href="registerSession.php">
click me</a>
 </body></html>
```

Server Side Dynamic Web Documents with PHP (10)

- □ **Database access**

```
<?
$conn = new COM("ADODB.Connection") or die("Cannot start ADO");
$conn->Open("DRIVER={Microsoft Access Driver (*.mdb)};
    DBQ=C:\\www\\database\\User.mdb");
// SQL statement to build recordset.
$rs = $conn->Execute("SELECT * FROM USERPASS");
echo "<p>Below is a list of UserName in the User.MDB database.</p>";
while (!$rs->EOF) {
    $fv = $rs->Fields(0); // Display values in field 0
    echo "UserName: ".$fv->value ."<br>" ;
    $rs->MoveNext();
}
$rs->Close(); $conn->Close();
?>
```

Server Side Dynamic Web Documents with JSP

□ **Example**

```
<html>
<body bgcolor="white">
<h1> Request Information </h1>
<font size="4">
JSP Request Method: <%= request.getMethod() %> <br>
Request URI: <%= request.getRequestURI() %> <br>
Request Protocol: <%= request.getProtocol() %> <br>
Servlet path: <%= request.getServletPath() %> <br>
Path info: <%= request.getPathInfo() %> <br>
Path translated: <%= request.getPathTranslated() %> <br>
</body>
</html>
```

Server Side Dynamic Web Documents with JSP (2)

- **Using existed Objects**
  - *request*: class *HttpServletRequest*
  - *response*: class *HttpServletResponse*
  - *out*: class *PrintWriter*
  - *session*: class *HttpSession*
  - *application:* class *ServletContext*
  - *config:* class ServletConfig

Server Side Dynamic Web Documents with JSP (3)

- JSP writing
  - Tag <% …. %>
    - <%= expression %>
    - <%! [type variable;]+ %>
    - <% Java code %>
    - <%-- comments --%>