

CISC1100: WEB Project

Fall 2014

Web Projects:
develop your own web pages and make
them interesting/useful ...

The Internet

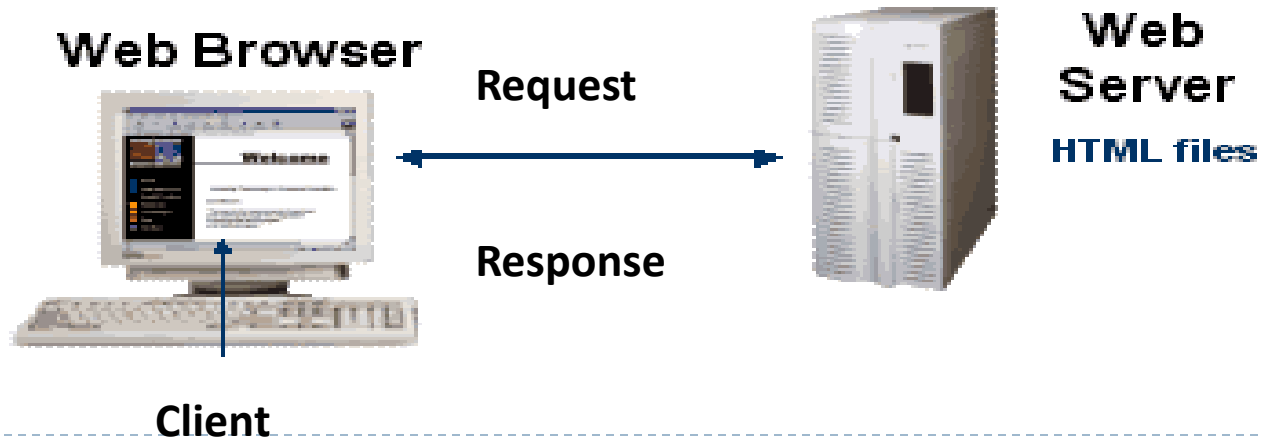
- A "network of networks" that consists of millions of smaller domestic, academic, business, and government networks.
 - Worldwide, publicly accessible
- Carrying information and services, such as **electronic mail, online chat, file transfer**, and the interlinked Web pages and other documents of the **World Wide Web**.

The World Wide Web

- ▶ Introduced in 1990 by Tim Berners-Lee
- ▶ A system of interlinked, **hypertext** documents accessed via the Internet.
 - ▶ With a web browser, a user views web pages that may contain **text, images, video, audio, and other multimedia**, and **navigates between them using hyperlinks**.

What happens behind the scene ?

- A web browser (**client**) lets a user request a resource.
- A web **server** processes client request and sends something back to the client.
- Clients and servers know HTML (**H**yper**T**ext **M**arkup **L**anguage), the agreed-upon standard for web pages



What is Web Browser ?

- ▶ **A Web browser** is a software that
 - ▶ enables a user to display and interact with the Web's rich multimedia content such as text, images, and other information.
 - ▶ The Web could be the World Wide Web, a local area network, or a web page on your own machine.
- ▶ **Microsoft Internet Explorer, Mozilla Firefox, Safari or Netscape Navigator**
 - ▶ The appearance of a Web page may differ between browsers.

URL & Hyperlinks

- ▶ URL (**U**niform/**U**niversal **R**esource **L**ocator)
 - ▶ Web page address – typing in Address field
 - ▶ HTTP (**H**yper**T**ext **T**ransfer **P**rotocol)
 - Protocol for transferring data over the Internet
 - ▶ HTTPS (**S**ecure **H**yper**T**ext **T**ransfer **P**rotocol)
 - Protocol for transferring encrypted data over the Internet.
 - ▶ Hyperlinks
 - ▶ Graphical or textual elements
 - ▶ Click to link to another Web page
 - ▶ Loads new page into browser window

Our environment: server

- ▶ **The server is storm.cis.fordham.edu**
 - ▶ An account with the server has been created for you (if you have requested for it)
 - ▶ You have an account name and password, also a file directory, email account, a URL

The client: OpenMail

- ▶ Our client software is OpenMail:
<http://storm.cis.fordham.edu/cgi-bin/openwebmail/openwebmail.pl>
- ▶ Using OpenMail, you can
 - ▶ Check emails (WebMail button)
 - ▶ View/Manipulate your directory (WebDisk)
 - ▶ Configure your account (e.g., change passwd, Pref)
- ▶ Detailed instruction at the course web site...
 - ▶ Create, edit a file, upload a file ...

Steps to build your web pages

- URL for your web page with storm is:
 - http://storm.cis.fordham.edu/~<your_account>
- When someone (anyone) enters above URL into web browser, browser will try to fetch & display
 - index.html file located under ~<your_account>/public_html/, at storm
- Therefore, you need to create a file named index.html under public_html directory

Build you home page ...

- ▶ Your index.html have to follow certain standard, **HTML** or **XHTML**
- ▶ Start with the simple example
 - ▶ copy and paste it to your editor
 - ▶ save and preview it
- ▶ Experiment by adding one thing a time
- ▶ Borrow someone else's good ideas
 - ▶ When you see a cool web page, select to view its source in the browser
 - ▶ Copy and Paste to your editor



HTML Primer

Basics of html

First HTML Example (1)

▶ Basic HTML Elements:

```
<html>
```

```
<head>
```

```
<title>My first Web Page</title>
```

```
</head>
```

```
<body bgcolor = "white">
```

```
<h2> My first Web Page</h2>
```

```
<p>I am so happy to be at Fordham!
```

```
</body>
```

```
</html>
```

head element

Head section: Title of the document and Style sheets and scripts

body element

Body section: Page's content the browser displays

What is HTML Element ?

- ▶ An HTML document is made up of various HTML elements
 - ▶ E.g., html element, head element, body element, img element...
- ▶ Every element starts with **Start tag** and ends with **End tag**, with element content in between, e.g.,

```
<head>
```

```
<title> This is my first webpage</title>
```

```
</head>
```

- ▶ Start tag may have **attributes** (provide additional information about an element)
 - ▶ A **name** and **value** pair specified in name=value format
 - ▶ e.g., to set background of web page to white:

```
<body bgcolor = "white">
```

```
<h1> My favorite band is </h1>
```

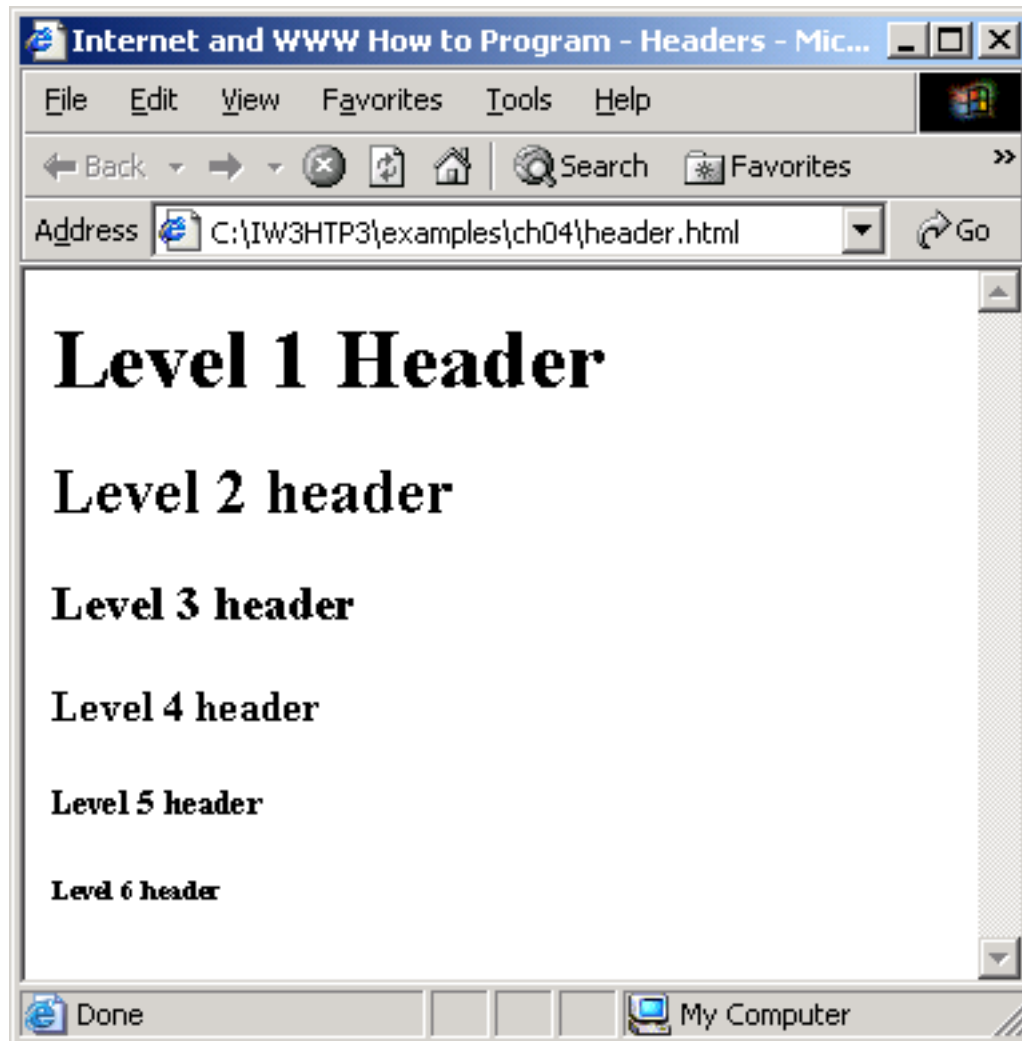
....



Headers Elements

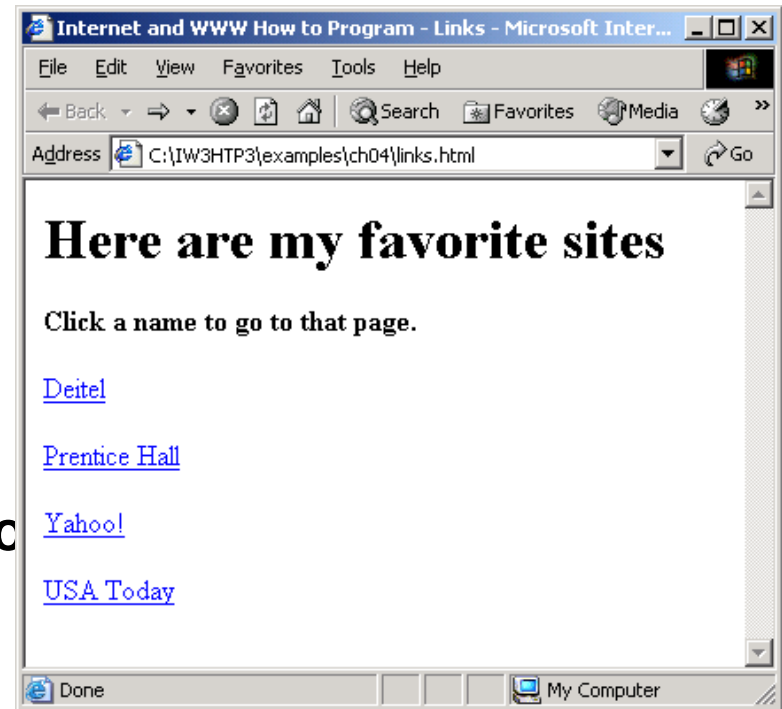
```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.4: header.html -->
6 <!-- XHTML headers      -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>Internet and WW How to Program - Headers</title>
11   </head>
12
13   <body>
14
15     <h1>Level 1 Header</h1>
16     <h2>Level 2 header</h2>
17     <h3>Level 3 header</h3>
18     <h4>Level 4 header</h4>
19     <h5>Level 5 header</h5>
20     <h6>Level 6 header</h6>
21
22   </body>
23 </html>
```





Adding HyperLink

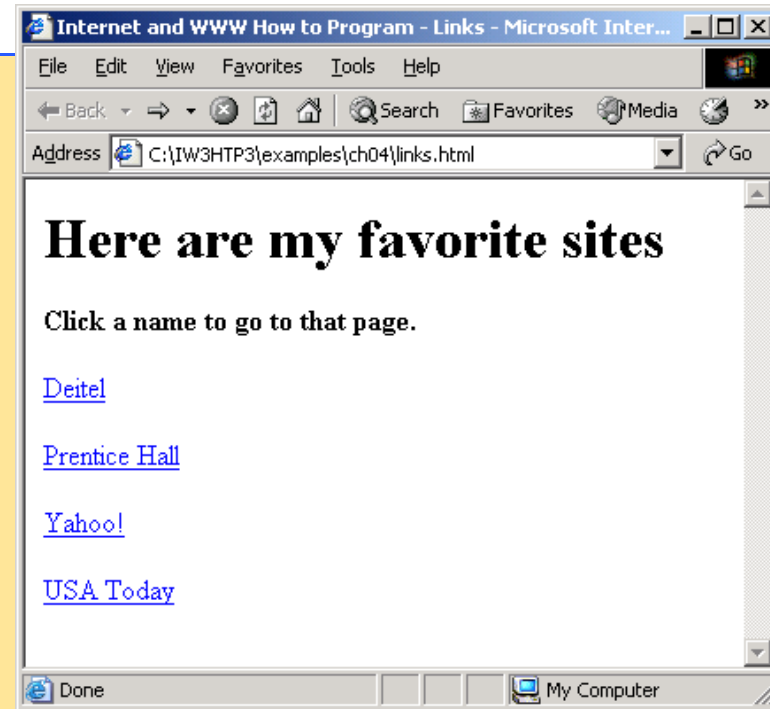
- ▶ References other sources such as HTML documents and images
 - ▶ Both text and images can act as hyperlinks
- ▶ Created using the **a (anchor) element**: `<a> `
 - ▶ Attribute `href` specifies the location of a linked resource :
`href = "http://www.yahoo.com"`
 - ▶ Link to e-mail addresses:
`href = "mailto:zhang@cis.fordham.edu"`



```

1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.5: links.html -->
6 <!-- Introduction to hyperlinks -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Links</title>
11  </head>
12
13  <body>
14
15    <h1>Here are my favorite sites</h1>
16
17    <p><strong>Click a name to go to that page.</strong></p>
18
19    <!-- Create four text hyperlinks -->
20    <p><a href = "http://www.deitel.com">Deitel</a></p>
21
22    <p><a href = "http://www.prenhall.com">Prentice Hall</a></p>
23
24    <p><a href = "http://www.yahoo.com">Yahoo!</a></p>
25

```



Paragraph element:
 <p> </p>

Start a new paragraph
 in the webpage



Adding Images Element (1)

- ▶ **img** element with attributes:
 - ▶ **src** attribute : Specifies the location of the image file
 - ▶ **width** and **height** attributes: **Pixels** (“picture elements”)
 - ▶ **alt** attribute : the text will be displayed if the browser could not display the image.
 - ▶ e.g.
``
- ▶ **Note:** img element is empty (nothing in between start and end tag)
 - ▶ Shorthand: terminated by character / before right angle bracket (>)
 - ▶ ``

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.7: picture.html -->
6 <!-- Adding images with XHTML -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>Internet and WWW How to Program - Welcome</title>
11   </head>
12
13   <body>
14
15     <p>
16       <img src = "xmlhttp.jpg" height = "238" width = "183"
17         alt = "XML How to Program book cover" />
18       <img src = "jhttp.jpg" height = "238" width = "183"
19         alt = "Java How to Program book cover" />
20     </p>
21   </body>
22 </html>
```



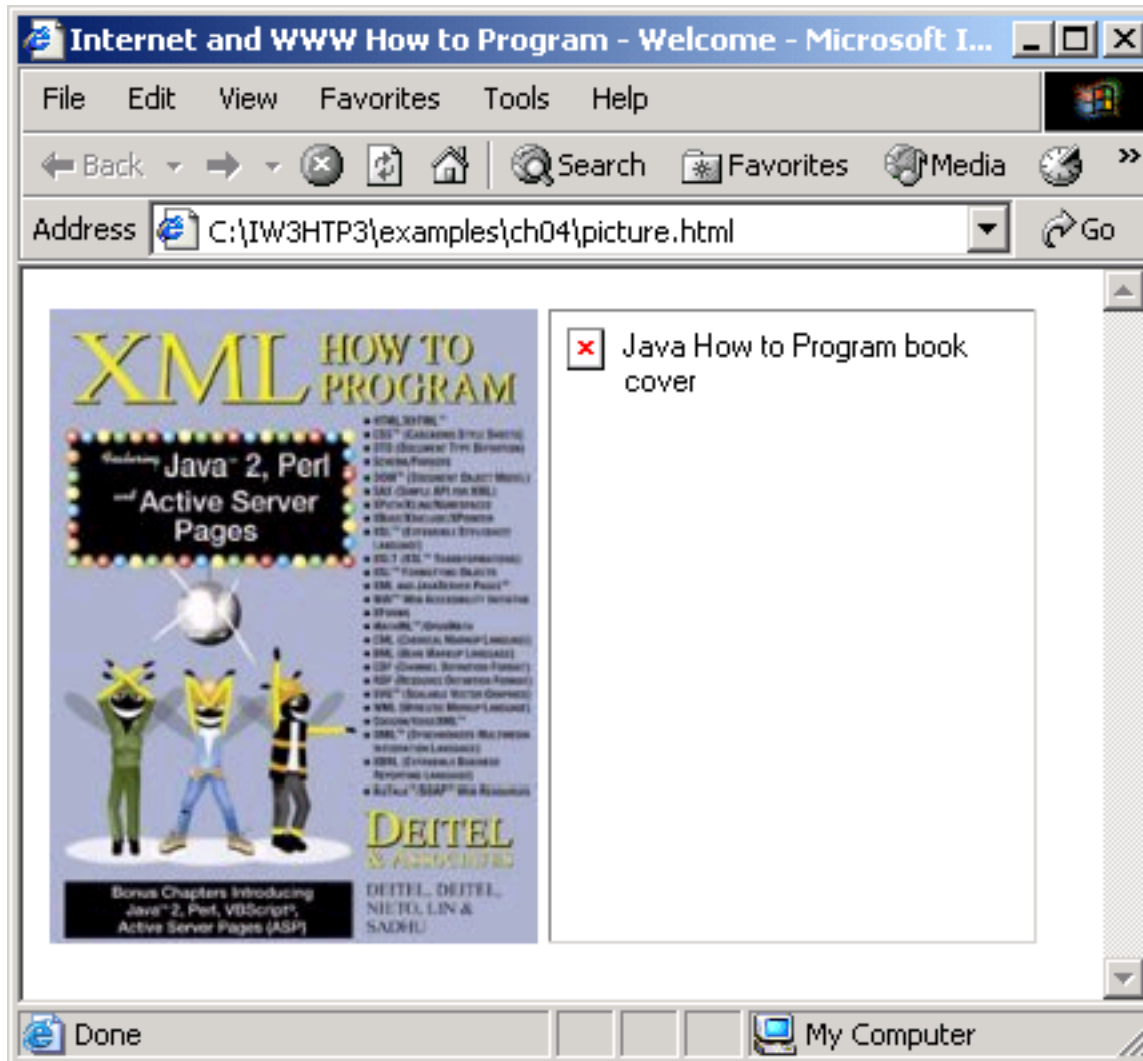


Image as Link

▶ Use an image as a link

```
<a href=http://www.yahoo.com>
```

```
  <img src= "yahoo.gif" width="232" height = "44"  
  /></a>
```

Internal Linking

- ▶ Enables user to jump between locations in the same document

- ▶ First, create an internal hyperlink destination by setting **attribute `id` of an element**

- ▶ Second, create an internal link to this element.

- ▶ Example:

```
<h1 id="Hobbies">Hobbies </h1>
```

:

```
<a href="#Hobbies">Find out my hobbies</a>
```

links (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 5.6: links.html -->
6 <!-- Internal Linking -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - List</title>
11  </head>
12
13  <body>
14
15    <!-- id attribute creates an internal hyperlink destination -->
16    <h1 id = "features">The Best Features of the Internet</h1>
17
18    <!-- an internal link's address is "#id" -->
19    <p><a href = "#bugs">Go to <em>Favorite Bugs</em></a></p>
20
21    <ul>
22      <li>You can meet people from countries
23        around the world.</li>
24
25      <li>You have access to new media as it becomes public:
```



links.html (3 of 3)

```
51     <li>Keeping in touch with old friends</li>
52     <li>It is the technology of the future!</li>
53 </ul>
54
55 <!-- id attribute creates an internal hyperlink destination -->
56 <h1 id = "bugs">My 3 Favorite Bugs</h1>
57
58 <p>
59
60     <!-- internal hyperlink to features -->
61     <a href = "#features">Go to <em>Favorite Features</em>
62     </a></p>
63
64 <ol>
65     <li>Fire Fly</li>
66     <li>Gal Ant</li>
67     <li>Roman Tic</li>
68 </ol>
69
70 </body>
71 </html>
```



Unordered Lists

- ▶ Unordered list element **ul**

- ▶ Creates a list in which each item begins with a bullet symbol (called a disc)
- ▶ **li** (list item): Entry in an unordered list
- ▶ Format:

```
<ul>  
  <li> first item </li>  
  <li> second item </li>  
</ul>
```

Nested and Ordered Lists

- ▶ Nested lists:
 - ▶ Represent hierarchical relationships
- ▶ Ordered lists (**o1**)
 - ▶ Creates a list in which **each item begins with a number**
 - ▶ Format

```
<ol>  
  <li> first item </li>  
  <li> second item </li>  
</ol>
```

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.11: list.html -->
6 <!-- Advanced Lists: nested and ordered -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10     <title>Internet and WWW How to Program - Lists</title>
11   </head>
12
13   <body>
14
15     <h1>The Best Features of the Internet</h1>
16
17     <!-- create an unordered list -->
18     <ul>
19       <li>You can meet new people from countries around
20         the world.</li>
21       <li>
22         You have access to new media as it becomes public:
23
```

list.1
(1 of



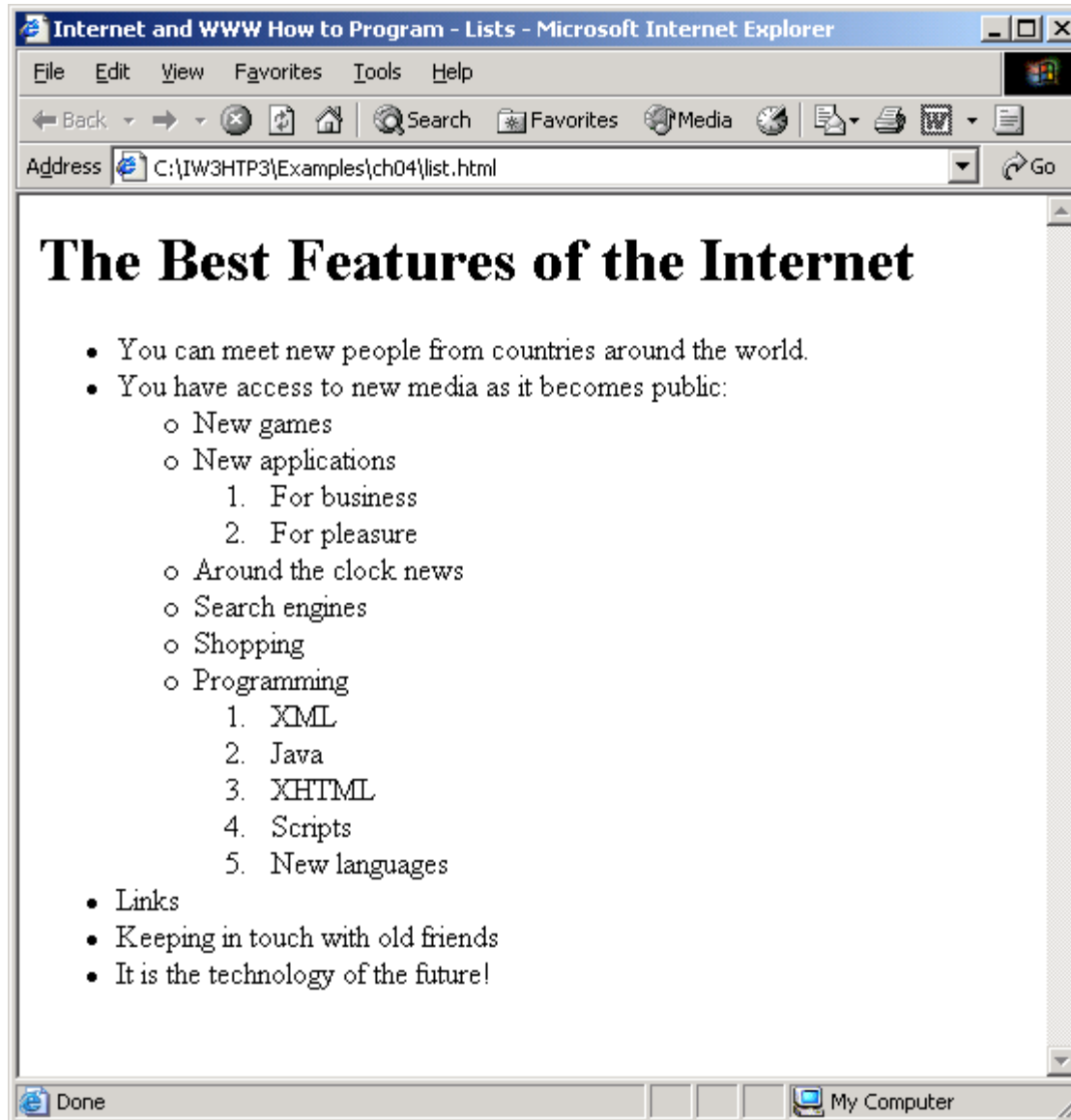
list.k (2 of

```
24 <!-- this starts a nested list, which uses a -->
25 <!-- modified bullet. The list ends when you -->
26 <!-- close the <ul> tag. -->
27 <ul>
28 <li>New games</li>
29 <li>
30 <ul>New applications
31
32 <!-- nested ordered list -->
33 <ol>
34 <li>For business</li>
35 <li>For pleasure</li>
36 </ol>
37 </li>
38
39 <li>Around the clock news</li>
40 <li>Search engines</li>
41 <li>Shopping</li>
42 <li>
43 <ul>Programming
44
45 <!-- another nested ordered list -->
46 <ol>
47 <li>XML</li>
48 <li>Java</li>
```

list.
(3 o

```
49         <li>XHTML</li>
50         <li>Scripts</li>
51         <li>New languages</li>
52     </ol>
53
54     </li>
55
56     </ul> <!-- ends the nested list of line 27 -->
57 </li>
58
59     <li>Links</li>
60     <li>Keeping in touch with old friends</li>
61     <li>It is the technology of the future!</li>
62
63 </ul> <!-- ends the unordered list of line 18 -->
64
65 </body>
66 </html>
```





So far, everything is static, i.e., the contents of the web page is fixed...

How to make the content adapts to the viewer, and the time ? Or make your web page interact with viewer ?

We need to do some simple programming ...

JavaScript

- ▶ **THE** scripting language for web
 - ▶ Enhances functionality and appearance of web pages
- ▶ **Client-side** scripting
 - ▶ The program is interpreted and executed at client side
 - ▶ Most web browsers support JavaScript
 - ▶ You could disable this on your browser

What can JavaScript do?

- ▶ a programming tool for Web designer
- ▶ put dynamic text into an HTML page
- ▶ react to events
 - ▶ e.g. when page finishes loading or when user click on HTML element
- ▶ read and write HTML elements
- ▶ validate data input in a form
 - ▶ before it is submitted to a server, saving server from extra processing
- ▶ detect the visitor's browser
 - ▶ To load page specifically designed for that browser
- ▶ create cookies
 - ▶ store and retrieve information on the visitor's computer

Simple JavaScript Program

- ▶ **Inline scripting**

- ▶ Written in the `<body>` or `<head>` of a document
- ▶ `<script>` tag
- ▶ Example: `<script type="text/javascript">`

`<!--`

script code here

`// -->`

`</script>`

JavaScript single-line
comment symbol: `//`

HTML comment tag, so that
browser doesn't show JavaScript
code in case JavaScript is not
supported

JavaScript objects

- ▶ JavaScript is an Object Oriented Programming (OOP) language
- ▶ Everything is an **object**
 - ▶ Each object is associated with some **attributes** (data) and behaviors (**methods**).
 - ▶ **Ex .**
 - ▶ **document object**: HTML document currently displayed
 - ▶ `writeln` method writes a line in the document
 - ▶ `document.writeln("<h1>welcome</h1>");`

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 7.1: welcome.html   -->
6 <!-- Displaying a line of text -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>A First Program in JavaScript</title>
11
12    <script type = "text/javascript">
13      <!--
14      document.writeln(
15        "<h1>welcome to JavaScript Programming!</h1>" );
16      // -->
17    </script>
18
19  </head><body></body>
20 </html>
```



Example One

Add this into the end of your <body> element:

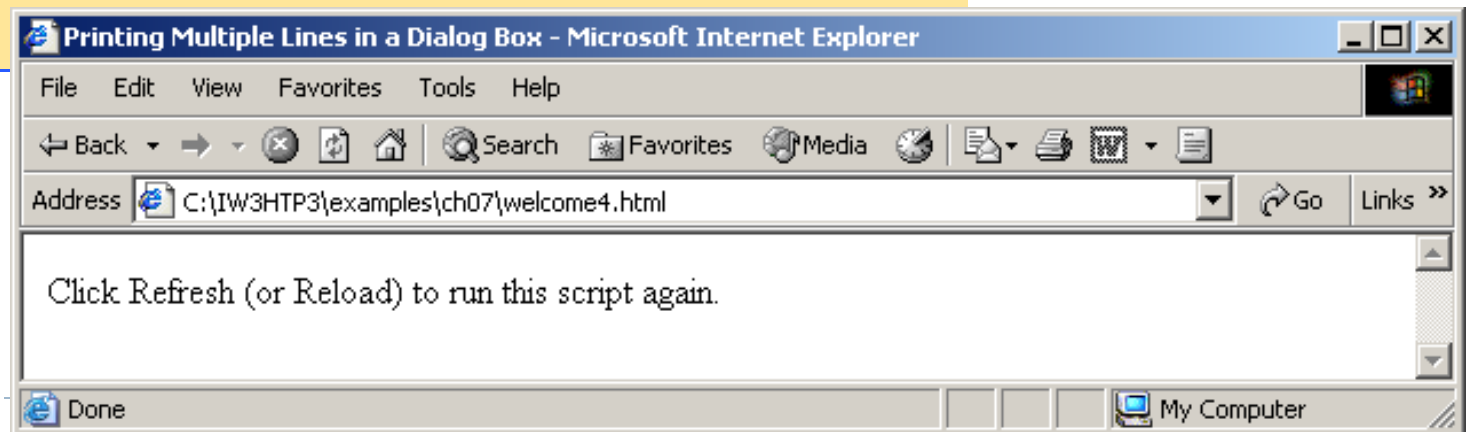
```
<script language="javascript"
    type="text/javascript">
<!--
    document.write("<p>Last updated on " +
        document.lastModified + "</p>");
-->
</script>
```

- ▶ A statement should end with a semicolon (;)
- ▶ JavaScript is **case sensitive**

JavaScript object: window

- ▶ **w**indow object : the browser window.
 - ▶ created automatically with every instance of a `<body>` or `<frameset>` tag
 - ▶ **a**lert method creates a Dialog box
 - ▶ `window.alert("welcome");`

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 7.4: welcome4.html          -->
6 <!-- Printing multiple lines in a dialog box -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head><title>Printing Multiple Lines in a Dialog Box</title>
10
11     <script type = "text/javascript">
12       <!--
13         window.alert( "Welcome to\nJavaScript\nProgramming!" );
14       // -->
15     </script>
16
17 </head>
18
19 <body>
20   <p>Click Refresh (or Reload) to run this script again.</p>
21 </body>
22 </html>
```



JavaScript: Variables

- ▶ **Variables are used to store data**
 - ▶ A variable's value can change during the script
 - ▶ You can refer to a variable by name to see its value or to change its value.
- ▶ **Name of a variable : a series of characters**
 - ▶ letters, digits, underscores(_) and dollar signs (\$)
 - ▶ no white space
 - ▶ not begin with a digit
 - ▶ not a keyword, word with a special meaning and reserved by language

Using Variables

- ▶ Declare a variable:
 - ▶ `var name;`
 - ▶ `var size;`
 - ▶ `var name, size;`
- ▶ Assign a value to a variable:
 - ▶ `name = "Lisa";`
 - ▶ `size = 20;`
- ▶ Combine two steps:
 - ▶ `var name = "Lisa";`

Read input from User

The image shows a dialog box titled 'User Prompt' with a close button (X) in the top right corner. The dialog contains the text 'Script Prompt: Please enter your name' and a text input field with the text 'GalAnt' inside. Below the text are two buttons: 'OK' and 'Cancel'. Four callout boxes with arrows point to different parts of the dialog: one to the prompt text, one to the text field, one to the OK button, and one to the default value 'GalAnt'.

This is the prompt to the user.

When the user clicks **OK**, the value typed by the user is returned to the program as a string.

This is the default value that appears when the dialog opens.

This is the text field in which the user types the value.

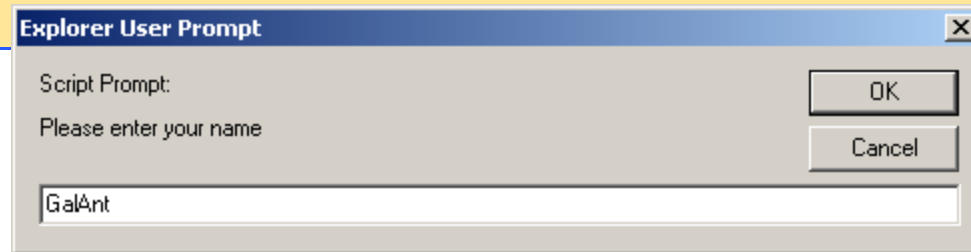
```
userName=window.prompt("Please enter your name", "GalAnt");
```

welcor (1 of 2)

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 7.6: welcome5.html -->
6 <!-- Using Prompt Boxes      -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Using Prompt and Alert Boxes</title>
11
12    <script type = "text/javascript">
13      <!--
14      var name; // string entered by the user
15
16      // read the name from the prompt box as a string
17      name = window.prompt( "Please enter your name", "GalAnt" );
18
19      document.writeln( "<h1>Hello, " + name +
20        ", welcome to JavaScript programming!</h1>" );
21      // -->
22    </script>
```



```
23
24 </head>
25
26 <body>
27 <p>Click Refresh (or Reload) to run this script again.</p>
28 </body>
29 </html>
```



Arithmetic Operation

- ▶ **Arithmetic operators:**
 - ▶ Addition: $\text{numberOne} + \text{numberTwo}$
 - ▶ Subtraction: $\text{numberOne} - 5$
 - ▶ Multiplication: $\text{numberOne} * \text{numberTwo}$
 - ▶ Division: $\text{numberTwo} / 6$
- ▶ **Rules of operator precedence**
 - ▶ $a - b * c$
 - ▶ $a * (b - c)$

Example: Adding Integers

- ▶ Prompt user for two integers and calculate the sum
- ▶ User input a string, i.e., a sequence of characters, need to be converted to number
 - ▶ `parseInt` method
 - ▶ Converts its string argument to an integer
 - ▶ `parseFloat` method
 - ▶ Converts its string argument to a float

```

1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5  <!-- Fig. 7.8: Addition.html -->
6  <!-- Addition Program      -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9     <head>
10        <title>An Addition Program</title>
11
12        <script type = "text/javascript">
13            <!--
14            var firstNumber,    // first string entered by user
15                secondNumber,  // second string entered by user
16                number1,       // first number to add
17                number2,       // second number to add
18                sum;           // sum of number1 and number2
19
20            // read in first number from user as a string
21            firstNumber =
22                window.prompt( "Enter first integer", "0" );
23

```

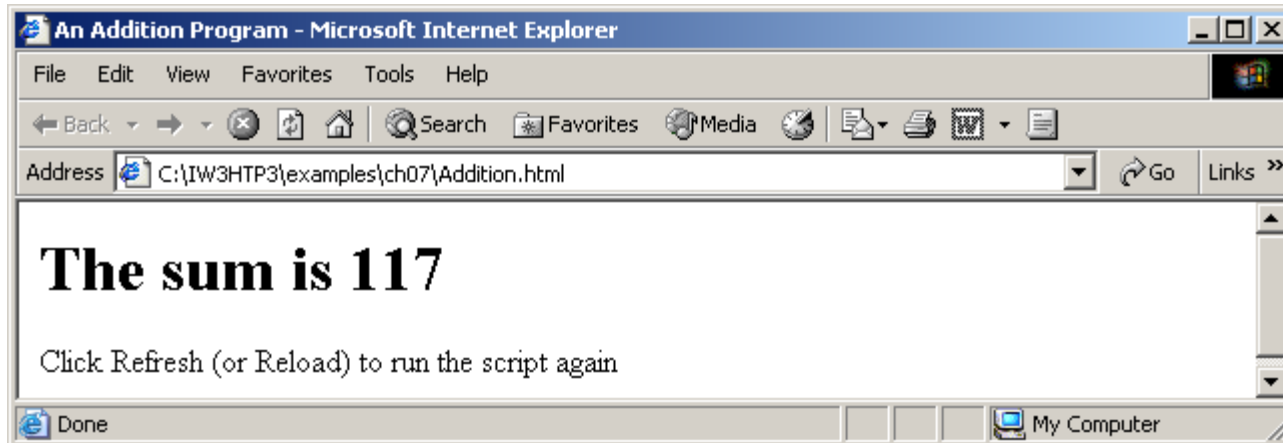
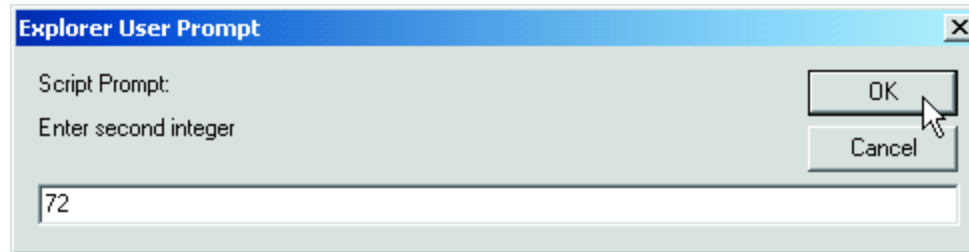
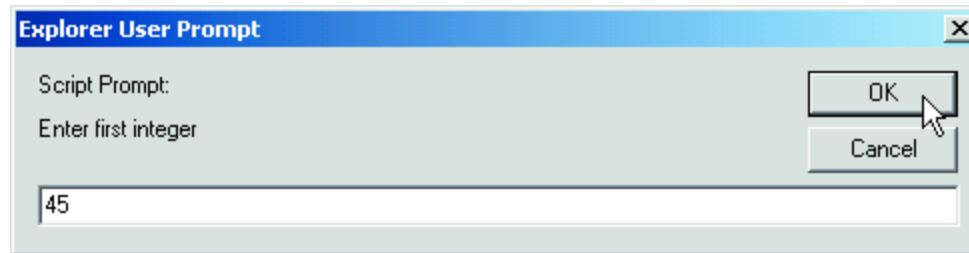
Add:
(1 of



Addit (2 of

```
24     // read in second number from user as a string
25     secondNumber =
26         window.prompt( "Enter second integer", "0" );
27
28     // convert numbers from strings to integers
29     number1 = parseInt( firstNumber );
30     number2 = parseInt( secondNumber );
31
32     // add the numbers
33     sum = number1 + number2;
34
35     // display the results
36     document.writeln( "<h1>The sum is " + sum + "</h1>" );
37     // -->
38 </script>
39
40 </head>
41 <body>
42     <p>Click Refresh (or Reload) to run the script again</p>
43 </body>
44 </html>
```





HTML Forms

- ▶ JavaScript use prompt window to ask for input
- ▶ Form: a vital tool to receive information from web surfer, `<form>... </form>`
 - ▶ their name, email address, credit card, etc.
 - ▶ See an example [here](#)
- ▶ And then,
 - ▶ store that data into a file
 - ▶ place an order
 - ▶ gather user statistics
 - ▶ register the person to your web forum...

Input fields in a form

- ▶ A HTML form is made up of input fields `<input>`
`...</input>`
- ▶ attributes of `<input>` :
 - ▶ *type* - what kind of input field: text, password, submit, radio, checkbox, ...
 - ▶ *name* - name of the field so that you may reference it later.
 - ▶ *size* - horizontal width , unit of measurement is in blank spaces.
 - ▶ *maxlength* - maximum number of characters that can be entered.

A quick tour of input fields

Form actions

- ▶ How to process the user inputs and invoke whatever need to be done ?
 - ▶ CGI: server side processing, transporting data to server to process
 - ▶ **JavaScript: implement simpler functions at the client side**

Form processing using JavaScript

- ▶ **Event**: user click on a button, user input some text into a text box, ...
- ▶ Register an **event handler** writing in JavaScript
 - ▶ which is invoked when the event happens, e.g., clicking a button.
- ▶ This [tutorial](#) explains how it works

A simple example

```
<HTML> <HEAD> <TITLE>Test Input</TITLE>
```

```
<SCRIPT LANGUAGE="JavaScript">
```

```
function testResults (form) {
```

```
    var TestVar = form.inputbox.value;
```

```
    alert ("You typed: " + TestVar);
```

```
}
```

```
</SCRIPT> </HEAD>
```

```
<BODY> <FORM NAME="myform" ACTION="" METHOD="GET">
```

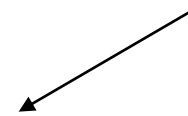
```
Enter something in the box: <BR> <INPUT TYPE="text" NAME="inputbox"  
    VALUE=""><P>
```

```
<INPUT TYPE="button" NAME="button" Value="Click"  
    onClick="testResults(this.form)">
```

```
</FORM> </BODY>
```

```
</HTML>
```

Important for server-side
Processing, where ACTION
attribute is set to URL of CGI



Write to an input field

```
<HTML> <HEAD> <TITLE>Test Input </TITLE>
<SCRIPT LANGUAGE="JavaScript">
function readText (form) {
    TestVar =form.inputbox.value;
    alert ("You typed: " + TestVar);
}
function writeText (form) {
    form.inputbox.value = "Have a nice day!"
} </SCRIPT> </HEAD>
<BODY> <FORM NAME="myform" ACTION="" METHOD="GET">
Enter something in the box: <BR> <INPUT TYPE="text" NAME="inputbox"
VALUE=""><P>
<INPUT TYPE="button" NAME="button1" Value="Read"
onClick="readText(this.form)">
<INPUT TYPE="button" NAME="button2" Value="Write"
onClick="writeText(this.form)"> </FORM> </BODY> </HTML>
```

Another example:
calculate square root

Part II of web project: Exploring JavaScript

- ▶ Extend your website, adding a form to your web page that invokes JavaScript to perform some task:
 - ▶ Math tools such as converter
 - ▶ Dice throwing experiment
 - ▶ Image slide show
 - ▶ Games

Part II of web project: Exploring JavaScript

- ▶ **Here are useful links**
 - ▶ [arithmetic in JavaScript](#)
 - ▶ [Javascript City](#)
 - ▶ [Java Script Kit](#)
 - ▶ [Free Javascripts](#)