Lecture Notes 2

Introduction to HTML and Cascaded Style Sheets (CSS)

Anand Tripathi

CSci 4131

Internet Programming

Topics of the Lecture

Refer to Chapter 2 of the textbook

- **Introduction to basic features of HTML**
  - **Notion of tag elements and attributes**
  - **Headers, fonts, and colors**
  - **Lists**
  - **Hypertext links**
  - **Images**
- **HTML5 Forms and Form Elements**
- **QUERY STRING and encoding format**

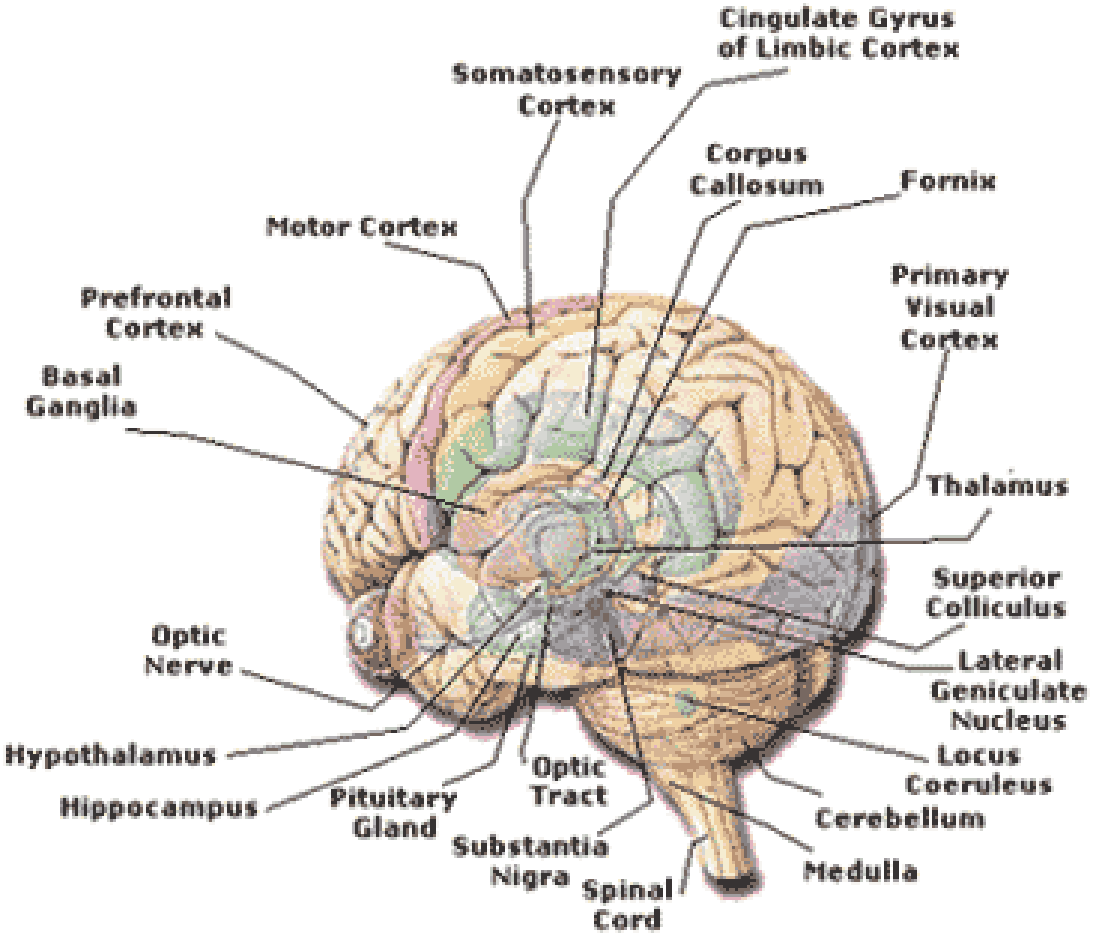
Introduction to XHTML and HTML5

- Originally HTML was based on SGML (Standard Generalized Markup Language). HTML version up to 4.0 were based on this.
  - Case-insensitive
- Later, XHTML was defined based on XML, which is a restrictive subset of SGML.
  - It requires documents to be well-formed.
  - It is **case-sensitive** – all tags are defined in *lowercase.*
- HTML5 provides interactive capabilities like additional form elements that can be syntactically checked, drag-and-drop, editing, and drawing when used in conjunction with JavaScript. Also supports video and audio objects in pages.

Basic Concepts in XML

- XML – eXtensible Markup Language - is a meta language for defining any desired markup language for an application domain.
  - XML has no tags of its own,
- An XML-based markup language can be used to store and communicate information.
- Using XML, new tags and structures can be defined to describe structure of the information to be stored and communicated.
- Newly created tags must adhere to the rules of XML specification.

Notion of Annotations and Markup



XML Example - Markup some text

Using XML we can assign some structure to the following text:

George

Washington

75

XML Example - Markup some text

```
<?xml version="1.0" encoding="UTF-8">

<person>
    <name>
        <firstname>    George        </firstname>
        <lastname>     Washington    </lastname>
    </name>
    <height unit = "inches" >    75    </height>
</person>
```

Element with tag "person"

Element with tag "firstname"

Element "height" has attribute named "unit" whose value is assigned as "inches"

Basic Concepts in XML

- *Elements* are the basic building blocks of an XML document.
- An element contains a *tag*, and possibly some *attributes*.
- A element may contain some *child elements*.

- A valid XML document must conform to certain structuring rules. The rules are defined either by a DTD (Document Type Definition) or an XML Schema.

Introduction to HTML

- Hypertext Mark-up Language (HTML): Documents can be "marked-up" for presentation and formatting and inclusion of "Hypertext" links.
- A Hypertext link in a document or object may point to a resource which may be present anywhere in the Internet.
  - Resource can be anything such as an HTML document, an executable program, a pdf document, an image, an audio file, a video file….
- HTML interpreters use the HTTP protocol for accessing a resource over the Internet.

HTML Features

- Hypertext links pointing to Internet resources.
- Presentation of text and images
  - Layout of images (size and position)
  - Control of font sizes, font faces, paragraphs, lists
  - Controlling color of fonts and background
  - Alignment control -- left, right, center of the page
  - Organization of information into tables
- Presentation of "forms" to users to submit/query information
- Embedding of executable code such as JavaScript to dynamically generate content or perform some processing on the client side.
- HTML is interpreted by the browser or any display utility.

HTML- Basic Building Blocks

- Document structure
  - XHTML (head and body part)
  - HTML5 (allows to create sections in the body)
- Lists
  - Ordered (numbered) and unordered (bulleted)
- Tables
  - XHTML defines table rows, and table data for each column in a row.
  - HTML5 defined table header and footer
- Forms
  - Various input data elements where user specifies data
  - HTML5 adds new elements that can be checked for syntax

HTML Elements -- Mark-up tags

- All formatting is controlled by mark-up tags, called elements.
- Any spacing, indentation, or other formatting in an input document has no significance in relation to output.
- A mark-up tag is enclosed in a pair of angular brackets
  < and >.
- Mark-up tags in XHTML are defined in lowercase.
- All tags are paired: such as
- to indicate start of centering
   to indicate end of centering

   a new paragraph
- A few tags are unpaired: For example:

 which starts a new line. This is an abbreviation for
</br>

GENERAL STUCTURE OF HTML DOCUMENT

```
<html>
   < head>
     <meta charset = “utf-8”>  </meta>
     < title> Document title < /title>
   </head>
  <body>
    here the contents of the document are described.
  </body>
</html>
```

Document Head and Body

- Document head part is enclosed in <head> element and it contains mostly meta data about the document, such as:
  - Document title (saved as bookmark by browsers)
  - Keywords and description for search engines
  - It can also contain JavaScript code (functions and data).

- Document text is enclosed in <body> element.
  - This contains the document contents such as text, images, hyperlinks.

Text Header Levels

Different levels of text heading levels can be used to emphasize the text.

<h1> defines largest size and bold-face

<h5> defines smallest size

See the next example.

```
<!DOCTYPE html>
<html>
<head>
   <title> An Example Page </title>
<head>
<body>
  <h1> This Page Illustrates headers  </h1>
  <h2> This Page Illustrates headers  </h2>
 <h3> This Page Illustrates headers  </h3>
 <h4> This Page Illustrates headers  </h4>
 <h5> This Page Illustrates headers  </h5>
 <h6> This Page Illustrates headers  </h6>
</body>
</html>
```

- Click here to see how the above html code gets displayed!

Ordered and Unordered Lists

- Ordered list has numbered items and it specified using <ol> tag
- Each element of list is enclosed in <li> tag

```
<ol>
    <li>  Item 1 </li>
    <li>  item 2 </li>
</ol>
```

- Unordered lists are specified using <ul> tag.
- A list can have a nested list.

Next example shows lists, and how to set background color and font color.

body and font Tags

```
<!DOCTYPE html>
<html>
<head>  <title> An Example Page 2 </title>    </head>
<body bgcolor="silver" vlink="red" link="blue">
 With the <body> tag one can specify attributes to control:
  <ol>
    <li> Background color using bgcolor    </li>
    <li> Hypertext link color using LINK      </li>
    <li> Visited Hypertext link color using VLINK </li>
  </ol>
 Colors in HEX:
  <ul>
     <li>  <font color="#FF0000"> RED = #FF0000          </font>   </li>
     <li> <font  color="#00FF00"> GREEN = #00FF00        </font>   </li>
      <li> <font  color="#0000FF"> BLUE =  #0000FF       </font>   </li>
     <li> <font  color="#FFFF00"> YELLOW =" #FFFF00     </font>   </li>
   </ul>
 The above text shows the of the tag to change font color. One can change font size and face.
</body>
</html>
```

Click here to see how the above html code gets displayed!

See example Fig 2-11 from the textbook

Images

HTTP proxy Servers

Click here to see the above image

ANCHORS AND HYPERTEXT LINKS

- Text or image objects can be marked to indicate them to represent Hypertext links.
- Anchor tag pair <a> and enclose a hyperlink.

- Example:
  One has to specify the URL of the object to which you wish this link to point.
- This anchor can be linked to a URL using the href attribute.
   Department Homepage

- Department Homepage

LINKING WITHIN A DOCUMENT

- Hypertext reference points can be created within a a large document using anchor tags.
- For example: In a large document divided into many sections, one can put a named anchor in the beginning of each section:
   <h1> Section 1: Introduction </h1>
- A Hypertext link can point to such any such anchor within the document.
  - For example: Anywhere in the document we can now put a Hypertext link pointing to the beginning of Section 1.
  - Click here to go to Section 1

    Section 1 of CSci 4131 Notes

How to create a web document?

- Log on a Unix/Linux machine in CSELab
- Your home directory will have a directory named ".www"
- Create an HTML document (file) with extension ".html" in the .www directory.
- Set permission for this file to 755
- If the name of your file is examle.html then can it can accessed using the following URL:
- http://www.cselabs.umn.edu/~YourLoginId/example.html

Tables

- Table is enclosed in <table> and </table> tags.
- Each row is enclosed in <tr> and </tr> tags
- A row may contain data for each column.
  - Table data for each column is contained within <td> and </td> tags.
- HTML5 has several other tags
  - <thead> for table header
  - <tbody> for table body
  - <tfoot> for table footer

TABLES - General Format

```
<table border=“5” cellspacing=“5” width="650>
<thead   bgcolor="#77DDDD">
    <th> Item Name</th>          <th> Price </th>      <th> Quantity</th>
</thead>
<tbody>
<tr>  <td> Apple iMac </td>      <td> $2000 </td>      <td> 5 </td>   </tr>
<tr>  <td> Dell Inspiron </td>   <td> $1000 </td>      <td> 5 </td>   </tr>
</tbody>
<tfoot>
       <tr> <td> Total </td> <td> $3000 </td>   <td> 10 </td>
</tfoot>
</table>
```

Click here to see this example

Book Example Fig 2.12 and 2.13

Figure 2-12 table also include <caption> tag for table caption.

```
<caption>
   <strong>Table of Fruits (1st column) and
      Their Prices (2nd column)</strong>
</caption>
```

See Book Example 2-12

Figure 2-13 show how data can be put spanning multiple columns or rows using attribute called “rowspan” or “columnspan”

See Book Example 2-12

Book Example Fig 2.13

```
<thead>
   <tr> <!-- merge two rows -->
       <th rowspan = "2">
           <img src = "camel.png" width = "205" height = "167"
                       alt = "Picture of a one-hump camel">
       </th>
       <th colspan = "4">
          <strong>Camelid comparison</strong><br/> Approximate as of 6/2011
       </th>
    </tr>
   <tr>
       <th># of humps</th>
       <th>Indigenous region</th>
       <th>Spits?</th>
       <th>Produces wool?</th>
  </tr>
</thead>
```

An Example of TABLES

```
<table border=5 cellspacing=5 WIDTH=650>
<tr bgcolor="#77DDDD">
   <th align="left">  Number  </th>
   <th>  Group / Individual  </th>
   <th> align="left" >  Topic </th>
   <th>  Handout Date </th>
   <th> Due Date </th>
</tr>
<tr bgoclor="#88EEEE">
   <td align="center">  1  </td>
   <td>  Individual   </td>
   <td> Programming of HTTP protocol using TCP/IP with C and UNIX </td>
   <td> <font color="RED">   September 29 </font>  </td>
   <td> <font color="RED"> October 8 </font> </td>
</tr>
</table>
```

See this example Table

- Click here to see how the above html code gets displayed!
- Click here to see border="0 and cellspacing="10
- Click here to see border="10 and cellspacing="0
- Click here to see border="0 and cellspacing="0

Special Symbols

In HTML5 specific strings are used to represent special symbols

Symbol	Symbol name	String representation in HTML
&	amperesand	&amp
’	apostrophe	&apos
>	greater	&gt
<	less	&lt
“	quote	&quot

See Fig 2-8 in the textbook for the listing of special symbols

FORMS

An html FORM is used to query a user and obtain input data.
Typical things one can do with a FORM are:

1. Ask the user to fill out certain input fields.
   E.g. - name, address etc on a FORM requiring personal information.
2. Checkout boxes for YES/NO kind of answers.
3. Radio buttons to present several choice, out of which one is to be selected.
4. Selection using scrolling options.
5. Input fields for PASSWORDS (does not display information typed in these fields).
6. RESET button.
7. SUBMIT button.
8. Fields to allow the user to input any arbitrary length text .

A simple example of a FORM

```
<!DOCTYPE html>
<html>
<body>
<form       method="GET"     action="echo.cgi">
    Name: <input   type="text"  name="firstname"  value="Enter your firstname" />
    <br/>
    Lastname: <input   type="text"  name="lastname"   value="Enter your lastname" />
   <br/>
   Age:  <input   type="text"   name="Age" />
   <br/>
   <input   type="SUBMIT"    value="SUBMIT FORM"/>
   <input   type="RESET"      value="CLEAR FORM TO START AGAIN "/>
</form>
</body>
</html>
```

Click here to see the effect of loading the above code.

Notice the following things

- When a request is submitted to the web server, the requested resource (program) specified in the action part is executed.
- In this example "echo.cgi" program is executed.
  - It is present in the same directory from where the HTML document containing the form was obtained.
- For the process executing this program echo.cgi, several environment variables are set by the web server.
  - HTTP_REFERER
  - QUERY_STRING
  - CONTENT_LENGTH
  - REQUEST_METHOD
  - REMOTE_ADDR

Environment Variables

environment variables are set as name=value pairs

SCRIPT_NAME = /~tripathi/Internet-Examples/XHTML/echo.cgi

SERVER_NAME = www-users.cs.umn.edu

HTTP_REFERER = http://www-users.cs.umn.edu/~tripathi/Internet-Examples/XHTML/sampleForm.html

REQUEST_METHOD = GET

SERVER_SOFTWARE = Apache

QUERY_STRING = firstname=Jack&lastname=Jumping&Age=5

REMOTE_PORT = 52224

SERVER_PORT = 80

REMOTE_ADDR = 128.101.35.7

SERVER_ADDR = 128.101.36.219

DOCUMENT_ROOT = /export/apache/null

HTTP_HOST = www-users.cs.umn.edu

Notice the following things

- The CGI program specified in the ACTION part receives a QUERY_STRING as a list of items name=value.
- The name=value items are separated by &
- Spaces are replaced by + character.

Control characters and non alphanumeric characters are replaced by % followed by 2 hexadecimal characters.

Method Name: Changing GET to POST has the following effect:

- QUERY_STRING is now empty.
- The input name=value pair is now available to the CGI program as standard input.
- Content_Length indicates the number of bytes available on the standard input stream.

Click here to see the execution of the same FORM with POST.

Input tag – type values

Input tag has type attribute which can be one of the following types:

- text
- password
- checkbox
- radio
- submit
- reset
- file
- hidden
- image
- button

Input tag – other attributes

Input tag can have several different additional attributes depending on the type:

type, name, value,

maxlength (applicable for certain types, such as text or file)

align, border,

checked, (for example radio or checkboxes)

onClick, onSelect, onChage, onFocus, onBlur,

Mostly used in conjunction with JavaScript

size (applicable to file, password, text)

Checkboxes

```
<!DOCTYPE html>
<html>
<body>
<form method="GET"   action="echo.cgi">
    Name: <input    type="text" name="firstname"   value="Enter your firstname" />
    <br/>
    Lastname: <input   type="text" name="lastname”   value="Enter your lastname"
     />
   <br/>
     Are you a smoker?   <input   type="CHECKBOX”   name="smoker“/>
   <br/>
     <input   type=" “SUBMIT” value="SUBMIT FORM">
     <input   type=" “RESET”  value="CLEAR FORM TO START AGAIN “/>
</form>
<body>  </html>
```

Click here to see the above FORM.

"RADIO" INPUTS

```
<form method="GET" action="echo.cgi">
      Name: <input type="text" name="firstname"    value="Enter your firstname" />
    <br/>
       Lastname: <input type="text" name="lastname"  value="Enter your lastname"
     />
    <br/>
       Are you a smoker?
       <input type="CHECKBOX”    name="smoker"  value="YES”    CHECKED />
    <br/>
     Are you alive?
           Yes       <input type="RADIO"  name="alive"   value="YES” CHECKED/>
           No        <input type="RADIO"  name="alive"  value="NO" />
           <input type="SUBMIT” value="SUBMIT FORM“ />
          <input type="RESET” value="CLEAR FORM TO START AGAIN “/>
</form>
```

Click here to see the above FORM code.

CHECKBOX vs. "RADIO" INPUT

```
<form method="GET"  action="echo.cgi">
   Name: <input type="text" name="firstname" value="Enter your firstname" />
   <br/>
   Lastname: <input type="text" name="lastname" value="Enter your lastname" />
   <br/> Are you a smoker?
    YES <input type="CHECKBOX" name="smoker" value="YES”   CHECKED>/
   "NO" <input type="CHECKBOX" name="smoker" value="NO" />
   <br/> Are you alive?
   Yes <input type="RADIO" name="alive" value="YES   CHECKED/>
   No <input type="RADIO" name="alive" value="NO" />
   <br/>
   <input  type="SUBMIT”  value="SUBMIT FORM“/>
   <input  type="RESET”   value="CLEAR FORM TO START AGAIN “/>
</form>
```

See effects of Checkbox vs. Radio inputs.

```
<form method="GET" action="echo.cgi“/>
   Name: <input  type="text" name="firstname" value="Enter your firstname" />
    <br/>
    Lastname: <input type="text" name="lastname" value="Enter your lastname"
    />
   <br/> Are you a smoker?
    YES <input type="CHECKBOX” name="smoker" value="YES”  CHECKED/>
    <br/>
    Are you alive?
    Yes <input type="RADIO" name="alive" value="YES/>
    <br/>
    <input type="SUBMIT” value="SUBMIT FORM“/>
    <input type="RESET” value="CLEAR FORM TO START AGAIN “/>
</form>
```

See the effect giving no choices in a radio **input.**

"RADIO" INPUT WITH MULTIPLE VALUES

```
<form method="GET"  action="echo.cgi“/>
   Are you alive?
   Yes <input   type="RADIO“ name="alive" value="YES” CHECKED />
   No  <input    type="RADIO“ name="alive" value="NO">
   Sometimes <input type="RADIO“ name="alive" value="MAYBE” />
   <br/>
   <input type="SUBMIT” value="SUBMIT FORM“/>
   <input type="RESET” value="CLEAR FORM TO START AGAIN “/>
</form>
```

See the effect multiple choices in a radio input.

SELECTION

```
<form method="GET"  action="echo.cgi">
  Please indicate your student status:
  <br/>
  <select name="studentStatus>
            <option value="ug-jr “> Junior Undergraduate </option>
            <option value="ug-sr” > Senior Undergraduate </option>
            <option value="grad-ms”> MS Graduate Student </option>
            <option value="grad-phd”> Ph.D. Graduate Student </option>
            <option value="ext”> Extension </option>
            <option value="special”>Adult Special </option>
            <option value="visitor”> Visitor </option>
  </select>
  <br/>
  <input type="SUBMIT” value="SUBMIT FORM">
  <input type="RESET” value="CLEAR FORM TO START AGAIN ">
</form>
```

Click here to see the above html code displayed.

SCROLLING WINDOW FOR SELECTION optionS

```
<form method="GET"  action="echo.cgi">
  Please indicate your student status:
  <br/>
  <SELECT name="studentStatus”  SIZE= “4”>
            <option value="ug-jr” > Junior Undergraduate </option>
            <option value="ug-sr” > Senior Undergraduate </option>
            <option value="grad-ms”> MS Graduate Student </option>
            <option value="grad-phd”> Ph.D. Graduate Student </option>
            <option value="ext”> Extension </option>
            <option value="special”>Adult Special </option>
            <option value="visitor”>  Visitor </option>
  </SELECT>
  <br/>
  <input type="SUBMIT” value="SUBMIT FORM">
  <input type="RESET” value="CLEAR FORM TO START AGAIN ">
</form>
```

Click here to see the above html code displayed.

MULTIPLE SELECTIONS

```
<form method="GET"  action="echo.cgi">
   Please indicate your favorite topics in computer science:
  <br/>
  <select    name="subjects”   multiple>
              <option value="os”> Operating System </option>
              <option value="internet”> Internet Programming </option>
              <option value="compls”> Compilers </option>
              <option value="arch”> Architecture </option>
              <option value="theory”> Theory of Computation </option>
              <option value="numbs”> Numerical Analysis </option>
              <option value="graphs”> Graphics </option>
              <option value="parallel”> Parallel Programming </option>
              <option value="dbms”> Databases </option>
              <option value="oo”> Object-Oriented Programming </option>
   </select>
   <br/> <input type="SUBMIT” value="SUBMIT FORM">
            <input type="RESET” value="CLEAR FORM TO START AGAIN ">
</form>
```

Click here to see the above html code displayed.

TEXT-AREAS

Use for getting any arbitrary length text input from the user.

```
<form method="GET"   action="echo.cgi">
   <center> <h3> Form to Fill-out Your Life-Story</h3>  </center>
   <textarea name="story" rows= "10 cols= "70" >
      Please enter here your life-story here.
   </textarea>
   <br/>
   <input type="SUBMIT" value="SUBMIT FORM">
   <input type="RESET"  value="CLEAR FORM TO START AGAIN ">
</form>
```

Click here to see the above html code displayed.

Form Input -- Hidden Field

- Can be used to include some information in a from which is not visible to the user.
- It is sent back to the server when a form is processed.
- It can be used for tracking activity or a user or relating a form submission to some previous activity by the user

Form Input -- Hidden Field

- Can be used to include some information in a form which is not visible to the user.
- It is sent back to the server when a form is processed.
- It can be used for tracking activity or a user or relating a form submission to some previous activity by the user

A simple example of a FORM

```
<body>
<form        method="GET”      action="echo.cgi”>
   <input type = "hidden"  name="tracking" value = "num5432" />
    Name: <input   type="text”  name="firstname”  value="Enter your firstname” />
    <br/>
    Lastname: <input   type="text”  name="lastname”   value="Enter your lastname” />
   <br/>
   Age:  <input   type="text”   name="Age” />
   <br/>
   <input   type="SUBMIT”    value="SUBMIT FORM”/>
   <input   type="RESET”      value="CLEAR FORM TO START AGAIN “/>
</form>
</body>
</html>
```

Click here to see the effect of loading the above code.

New Form Elements Introduced in HTML5

HTML5 introduced several new form elements that allow browser to check if the user has entered the form data in the correct form:

- email
- tel for telephone numbers
- date
- datetime
- number

See example 3.1 from the textbook

Proper HTML5 Document Structure

- Elements must be **properly nested**
- Elements must always be **closed**

 First paragraph

 Second paragraph

Above is wrong; correct way

 First paragraph

 Second paragraph

- Elements must be in **lowercase**
- Documents must have **one root element**

Common errors

- Some of the most common errors in the usage of XHTML are:
- Not closing empty elements (elements without closing tags in html4)
  - Incorrect:
  - **Correct:
**
    Note that any of these are acceptable in HTML:
</br>, and
. Older HTML-only browsers interpreting it as HTML will generally accept
 and
.
- Not closing non-empty elements
  - Incorrect: This is a paragraph.
    This is another paragraph.
  - **Correct:**
    ** This is a paragraph. **
    ** This is another paragraph. **

Common errors

- Improperly nesting elements
  - Incorrect:

    This is some text.
  - Correct:

    This is some text.
- Not putting quotation marks around attribute values
  - Incorrect:

    <td rowspan=3>
  - Correct: <td rowspan="3">

Common errors

- Using the ampersand character outside of entities (Note that this would also be invalid in html)
  - Incorrect: <title>Cars & trucks</title>
  - Correct: <title>Cars & trucks</title>
  - Incorrect: News
  - Correct: News
- Failing to recognize that XHTML elements and attributes are case sensitive
  - Incorrect: <body> The Best Page Ever </body>
  - Correct: <body> The Best Page Ever </body>
- Using attribute minimization
  - Incorrect: <textarea readonly>Read-ONLY</textarea>
  - Correct: <textarea readonly="readonly">Read-ONLY</textarea>

Cascading Style Sheets

- See Deitel's book online: Chapter 5 for examples
- Principle of separation of concerns:
  - It separates the contents of a document from the directives for its layout and presentation style,
  - Layout and presentation info is called "style"
  - For the same content, we can apply a different style to get a different presentation view
  - A style can be attached to an element individually or one can specify a style to be applied to all elements of a type in a document.
  - One can define a style and apply it selectively to some elements. Class

Style Sheets

- Style sheets allow controlling the presentation style for a document.
  - For example, one can say that all elements with a certain tag name should be displayed in a given manner.
  - For example one can change the top-level headers to be presented in bold and “red” color font.
- Style commands can be either contained in the document file itself, or they may be defined in separate files for import by any document.

Example of Embedded Styles

Example 5.4

```
<head>
    <title>Embedded Styles</title>
    <style type =" "text/css">
      body     { font-family: arial, helvetica, sans-serif }
      p        {font-size: 12 pt}  /* all paragraphs 12 pt font-size  */
      a.nodec  { text-decoration: none }
      a:hover  { text-decoration: underline } /* mouse over anchor element */
      li em    { font-weight: bold }  /* all nested em children of li should be bold */
      h1, em   { text-decoration: underline }  /*all  h1 and em elements underlined*/
      ul       { margin-left: 20px }  /*all  ul elements set at left margin 20 pixels */
      ul ul    { font-size: .8em; color: red }
            /* all nested unnumbered list element should be in red color */
      .special { color: blue }    /* define a class named special  */
    </style>
  </head>
```

```
<body>
    <h1> List of Topics  </h1>
    <ul>
      <li> Client-Server Model </li>
      <li> Basics of Web Technologies
        <ul>
          <li> http Protocol </li>
          <li>  XHTML </li>
          <li>  Forms </li>
          <li>  Style Sheets </li>
        </ul>
      </li>
      <li> Web Programming <em> using Java </em>  </li>
      <li> Javascript </li>
      <li> CGI Programming <em> using Perl </em></li>
    </ul>
    <p><em>Go to the class webpage for </em>
      <a class ="nodec" href =" "http://www.cs.umn.edu">
        CSci 5131:  Advanced Internet Programming </a>
    </p>
    <p class="special> Check again more description of topics. </p>
  </body>
```

Click here to see this document

Style Definitions

.special defines a “class”

An element with class attribute set equal to “special” is displayed with that style.

a.nodec defines class which can only be applied to anchor elements

hover is a pseudo class, this class is activated when mouse is put over an element; In this example it is associated with anchor elements

#elementIdName will be applied to the specific element whose id=“elementIdName”

Example of Linking External Style Sheets

```
<html xmlns ="http://www.w3.org/1999/xhtml">
  <head>
    <title>Linking External Style Sheets</title>
    <link rel ="stylesheet" type ="text/css"
      href ="styles.css" />
  </head>
  <body>  …     </body>
  </html>
```

Click here to see the example

Style File: styles.css

```
/*  External style file example */
        body     { font-family: arial, helvetica, sans-serif }
        p        {font-size: 12 pt}
        a.nodec  { text-decoration: none }
        a:hover  { text-decoration: underline }
        li em    { font-weight: bold }
        h1, em   { text-decoration: underline }
        ul       { margin-left: 20px }
        ul ul    { font-size: .8em; color: red }
        .special { color: blue }
```

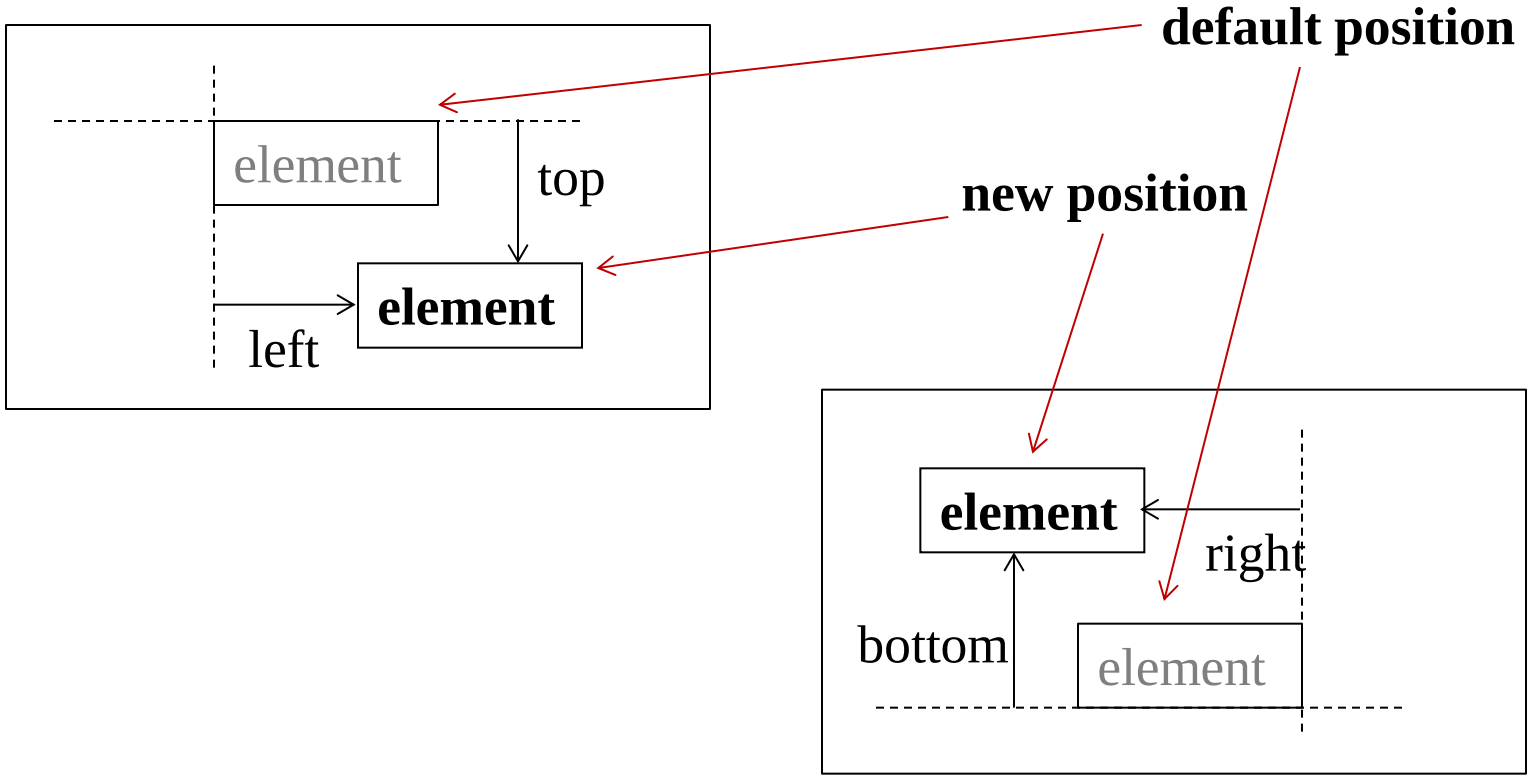
Example of Positioning and Superimposing Images
See Book example Chapter 5 – Figure 6

```
<head>
    <title>Absolute Positioning</title>
    <style type =" "text/css">
    .bgimg   { position: absolute;
               top: 0px;
               left: 0px;
               z-index: 1 }
    .fgimg   { position: absolute;
               top: 25px;
               left: 100px;
               z-index: 2 }
    .text    { position: absolute;
               top: 50px;
               left: 200px;
               z-index: 3;
               font-size: 20pt;
               font-family: tahoma, geneva, sans-serif }
    </style>
  </head>
```

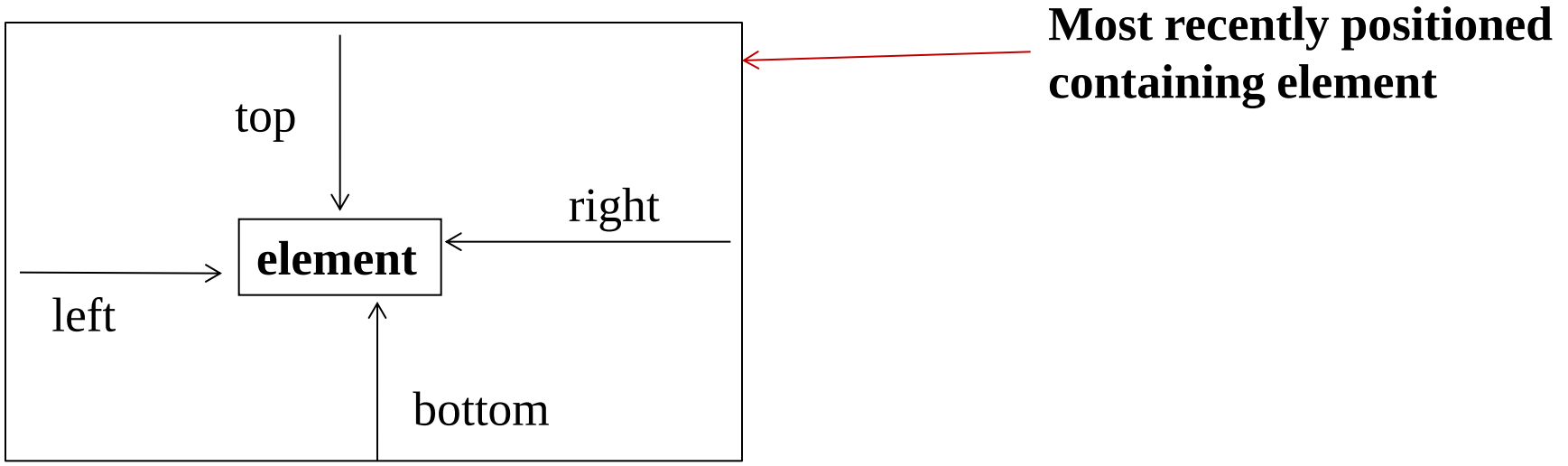
Position property

- position: static
  - By default - positioned according to HTML document flow
- position: relative
  - Positioned relative to its normal position. It is moved relative to the specified “side:
    - **top**
    - **bottom**
    - **left**
    - **right**
- position: absolute
  - Absolute w.r.t most recently positioned parent element
- position: fixed
  - Element stays at the same place in viewport irrespective of scrolling

Relative Positioning – two illustrations



Absolute Positioning



Example of Positioning and Superimposing Images
See Book example Chapter 5 – Figure 6

```
<head>
    <title>Absolute Positioning</title>
    <style type =" "text/css">
    .bgimg   { position: absolute;
               top: 0px;
               left: 0px;
               z-index: 1 }
    .fgimg   { position: absolute;
               top: 25px;
               left: 100px;
               z-index: 2 }
    .text    { position: absolute;
               top: 50px;
               left: 200px;
               z-index: 3;
               font-size: 20pt;
               font-family: tahoma, geneva, sans-serif }
    </style>
  </head>
```

```
<body>
    <p><img src ="bgimg.gif" class ="bgimg"
      alt =" "First positioned image" /></p>

    <p><img src ="fgimg.gif" class ="fgimg"
      alt ="Second positioned image" /></p>

    <p class ="text">Positioned Text</p>
  </body>
```

Click here to see this example

Example of Positioning Text

```
<style type =" "text/css">
        p           { font-size: 1.3em;
                      font-family: verdana, arial, sans-serif }
        span         { color: red;
                      font-size: .6em;
                      height: 1em }
        .super       { position: relative;
                      top: -1ex }
        .sub         { position: relative;
                      bottom: -1ex }
        .shiftleft   { position: relative;
                      left: -1ex }
        .shiftright { position: relative;
                      right: -1ex
</style>
```

Comment 1 em is the font-size.

The ‘ex’ unit is defined as the height of the lowercase "x"

Example of Positioning Text

```
<body>
    <p>The text at the end of this sentence
    <span class ="super">is in superscript</span>.</p>
    <p>The text at the end of this sentence
    <span class ="sub">is in subscript</span>.</p>
    <p>The text at the end of this sentence
    <span class ="shiftleft">is shifted left</span>.</p>
    <p>The text at the end of this sentence
    <span class ="shiftright">is shifted right</span>.</p>
  </body>
```

Click here to see this example

div, span, nav tags

- div
  - It is used to group a set of elements so a style could be applied to them as a whole.
- span
  - It is used to enclose some text (say within a paragraph) so some style can be applied to that text.
- nav
  - It is used to enclosed important navigation links in a document

section and meta tags

- HTML5 introduced section tag
- A document can structured into different sections, and each section is enclosed in a section tag

- meta tags to include information about the document such as keywords and description, to be used by crawlers

  <meta name = “keyword” content = “…. “ />

  <meta name = “description” content = “…. “ />

Example of CSS with floats

```
<html> <head>
<title>Two columns with CSS</title>
<style>
.left    {width: 200px;  float: left;  padding: 5px;  background: yellow;
}
.right   {width: 600px;  padding: 5px;  margin-left: 205px;  background: lightgrey;
}
</style>
</head>

<body>
<div class="left"> Left area </div>
<div class="right"> Right area </div>
</body> </html>
```

- See this example here

Classpage using CSS

The following example includes "Server-side Include" features.

This file is stored as index.shtml (to indicate that it has server-side include features.

```
<html> <head>
<title>Two columns with CSS</title>
<style>
#left  { width: 200px; float: left; padding: 5px; background: darkblue; }
#right { padding: 5px; margin-left: 210px; background: lightgrey; }
</style>
</head>

<body>
<div id="left"> <!--#include file="menu.html"--> </div>
<div id="right"> <!--#include file="class_info.html"--> </div>
</body>
</html>
```

Click here to see this (The website presented in this example is not complete.)

Background Images

- Multiple background images can be specified.
- A background image can be set with options:
  - repeat
  - no-repeat
  - Position of the image

Background Images -repeat

Example of no-repeat option

We have two images:

```
#banner { width: 915px; font-family: "Helvetica", "Arial", "sans-serif";
    background-image: url(img9.gif), url(minneapolisSkyline-3.jpeg) ;
    background-position: top left, top left;
    background-repeat: no-repeat, no-repeat;
    background-origin: border-box; }

<body  id="banner">
  …..
</body>
```

Background Images -repeat

Example of with repeat option for skyline image

We have two images:

```
#banner { width: 915px; font-family: "Helvetica", "Arial", "sans-serif";
    background-image: url(img9.gif), url(minneapolisSkyline-3.jpeg) ;
    background-position: top left, top left;
    background-repeat: no-repeat, repeat;
    background-origin: border-box; }

<body  id=“banner”>
  …..
</body>
```

How create web documents on CSELab server?

- In your home directory you will find a directory named

  .www
- Set permissions on this directory to “711”

  chmod 711 .www
- Put the contents that you want to publish in the .www directory.
- Permissions for the contents should be set to 755
- If your login id is “student”, and you put “exam.html” in the .www directory, the URL will be
- http://www.cselabs.umn.edu/~student/exam.html