Web Design & Development

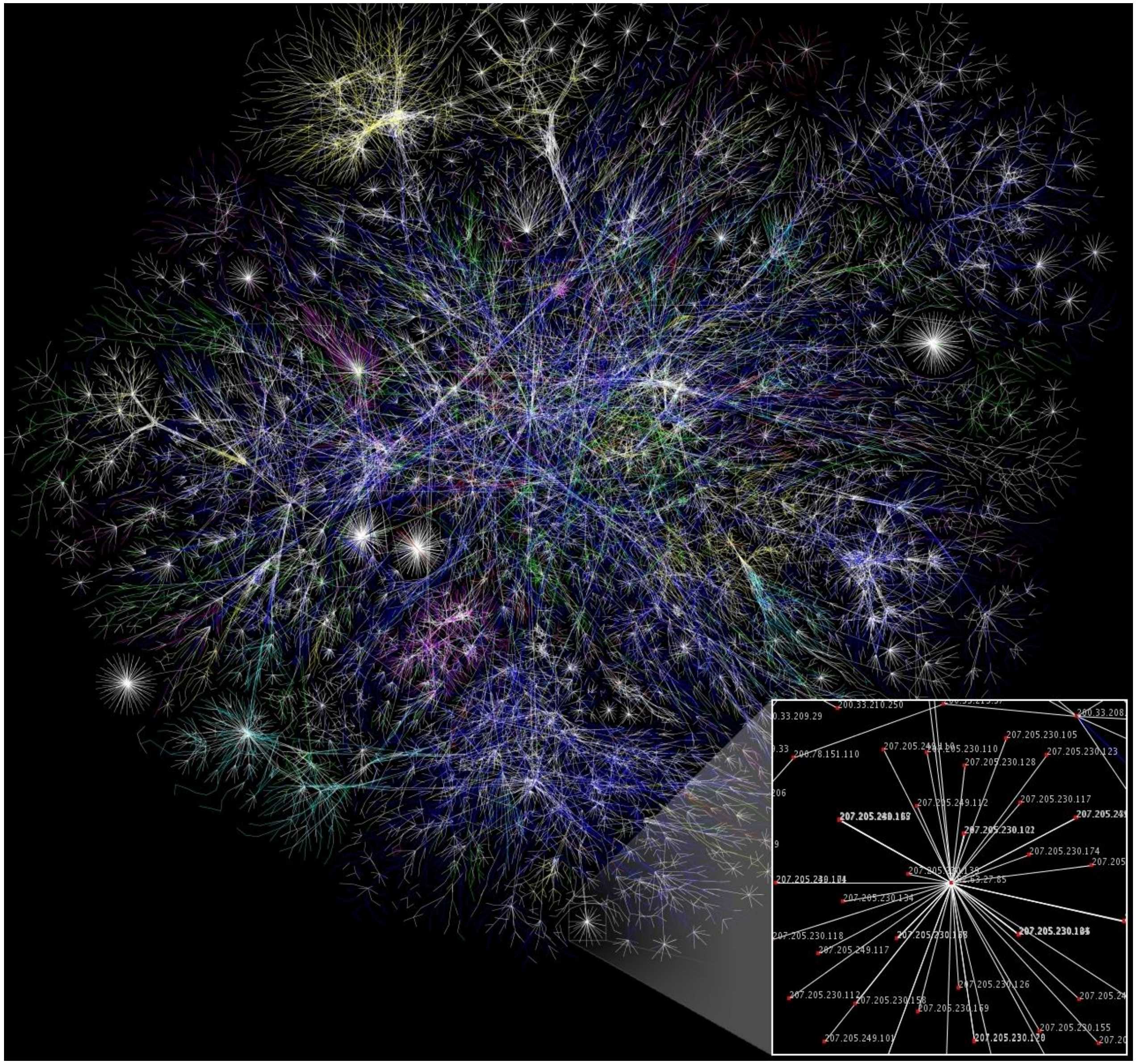
Introduction to the Internet

Mehmud Abliz

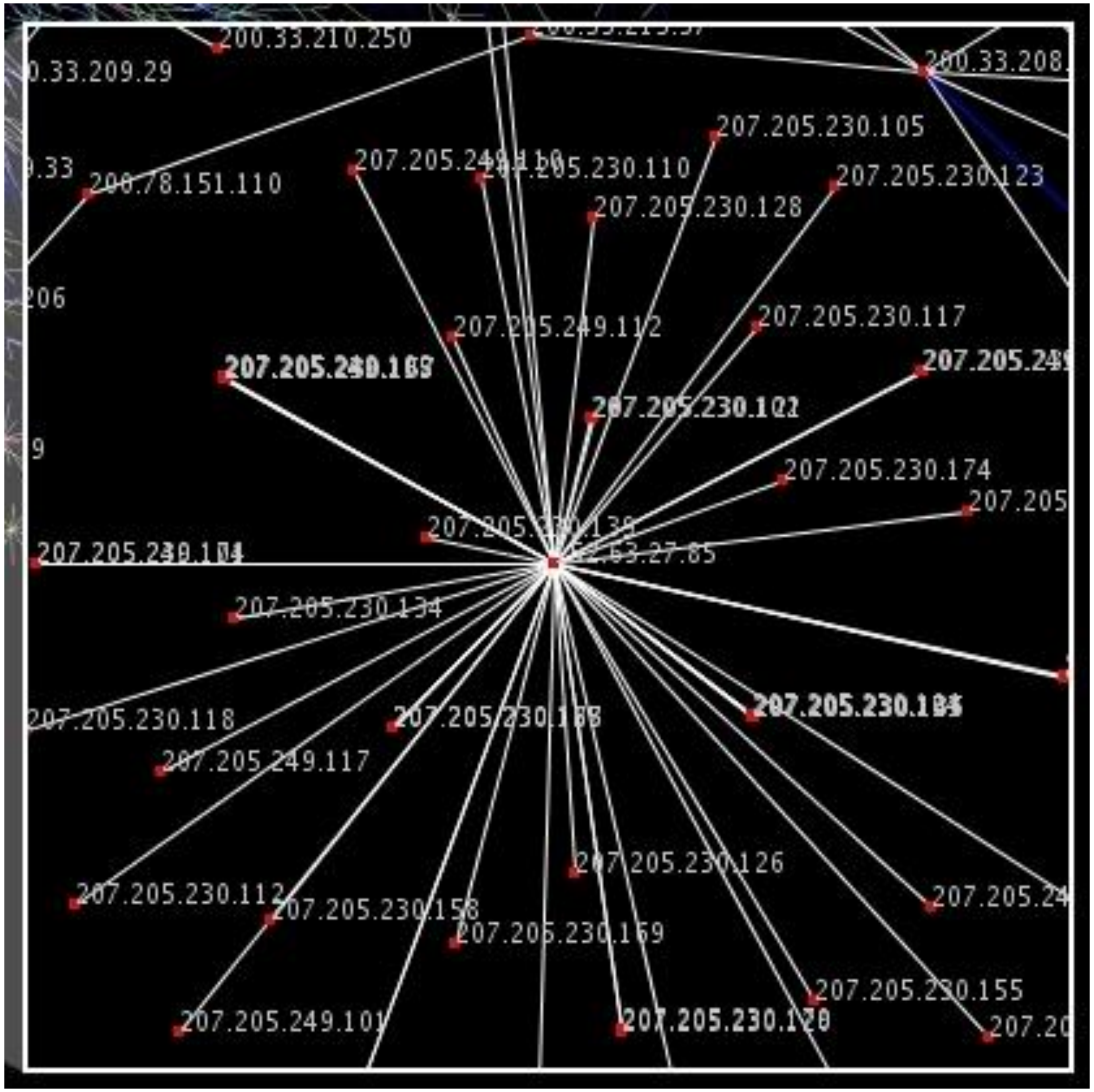
Internet, Packets and Routing

- Internet is a network of computer networks
- Data is transmitted by packet switching using the standard **Internet Protocol** (**IP**)
- **Packet** – a unit of information carriage
- **Packet switching** – process of moving packets from one node (computer device) to another

A Visualization of Internet



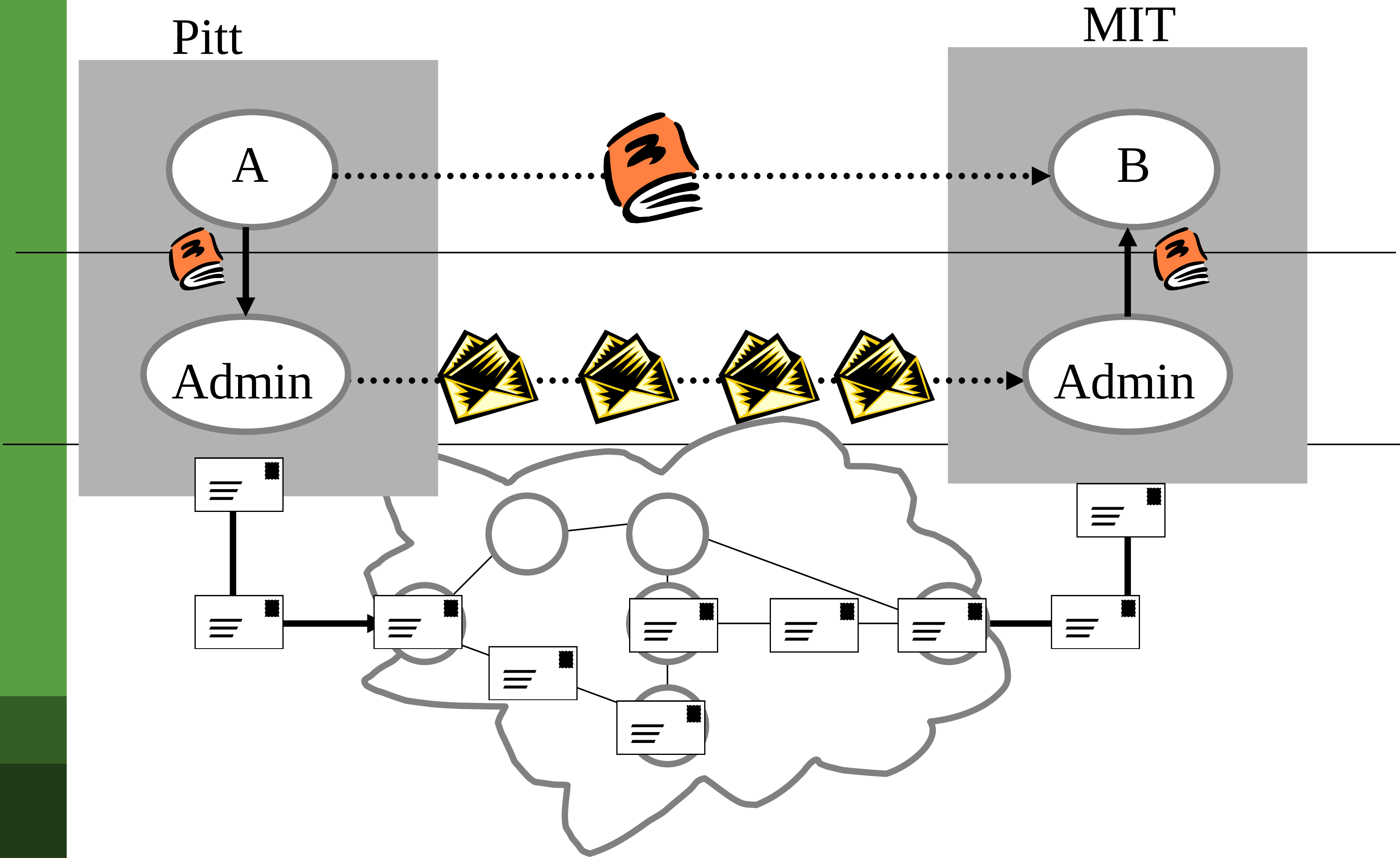
A Visualization of Internet



Internet, Packets and Routing

- At the sender, data is **broken into packets** and sent to the nearest node (**router**)
- At each router, it sends the packet to another router that is closer to the final destination
- At the receiver, packets are **reassembled** to get the original data
- A simple analogy: mailing system

Mailing System



TCP/IP and Domain Names

- Basic task of IP – moving packets as quickly as possible from one router to another
- Yet, it doesn’t check whether packets are delivered successfully, thus need **TCP**
- **TCP** (**Transmission Control Protocol**) – disassemble/reassemble packets, error checking, ACK packets

TCP/IP and Domain Names

- We need some sort of address in order to identify different nodes, as if every house has a mailing address in order to receive mail from others
- The one used by Internet Protocol is called **IP address**
- Every host on the Internet has a unique IP address, made up of four numbers. E.g.. 192.56.215.131, each number is between 0 and 255

TCP/IP and Domain Names

- The numbers in an IP address is hard to remember, while names are easier
- **Domain Name System** – a mapping between the human-readable name (domain name) of a host and its IP address
- A **domain name** consists of two or more parts, e.g. *cs.pitt.edu*
- The rightmost label conveys the top-level domain, e.g. *edu*

TCP/IP and Domain Names

- Each label to the left specifies a subdomain, in our example, subdomain is *pitt* (University of Pittsburgh), and sub-subdomain is *cs* (computer science).
- A top-level domain contains of multiple subdomains, each subdomain can contain multiple sub-subdomain, so on.
- The database contains the mapping between a domain name and an IP address is stored on a **DNS server**.

World Wide Web

- The **World Wide Web** (commonly shortened to the **Web**) is a system of interlinked, hypertext documents accessed via the Internet.
- It is created to share files/documents and overcome the barrier of different file formats
- **Hypertext** refers to text on a computer that will lead the user to other, related information on demand.

World Wide Web

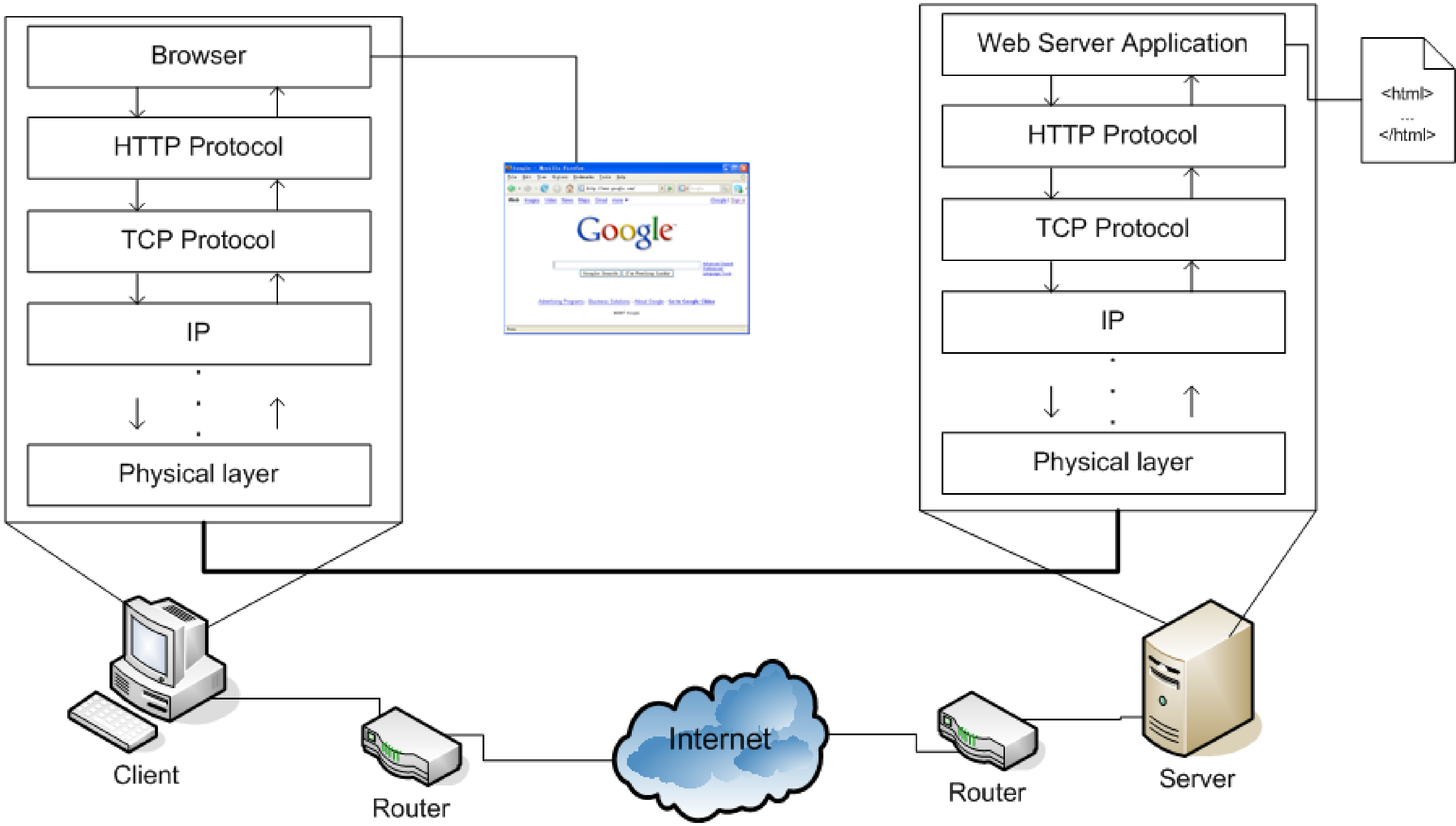
- hypertext documents are created using a special kind of document formatting or “markup” language called **HyperText Markup Language** (**HTML**).
- HTML is sent or received over the network using **HyperText Transfer Protocol** (**HTTP**).
- A **browser** is a software program which interprets the HTML documents and displays it on the user’s screen.

```
<!DOCTYPE html PUBLI
<html>
<!-- created 2003-12-12-->
 <head><title>XYZ</title>
 </head>
 <body>
 <p>
  voluptatem accusantium do
  totam rem aperiam eaque
 </p>
 </body>
</html>
```

- Each document/resource on the WWW needs to have an identifier in order to be accessed by others.
- A **Uniform Resource Identifier** (**URI**), is a compact string of characters used to identify or name a resource.
- A **Uniform Resource Locator** (**URL**) is a URI which provides means of obtaining the resource by describing its network "location".

- Two things are given by the URL
  - Exact location of the document
  - The method or protocol by which to retrieve and display the document
- Example, http://www.cs.pitt.edu/~mehmud/cs134/index.html
  - http:// – specifies the protocol
  - www.cs.pitt.edu – specifies the host name / domain name
  - /~mehmud/cs134/index.html – specifies the path of the document on the host

Putting it All Together



Creating a Basic Web Page

Mehmud Abliz

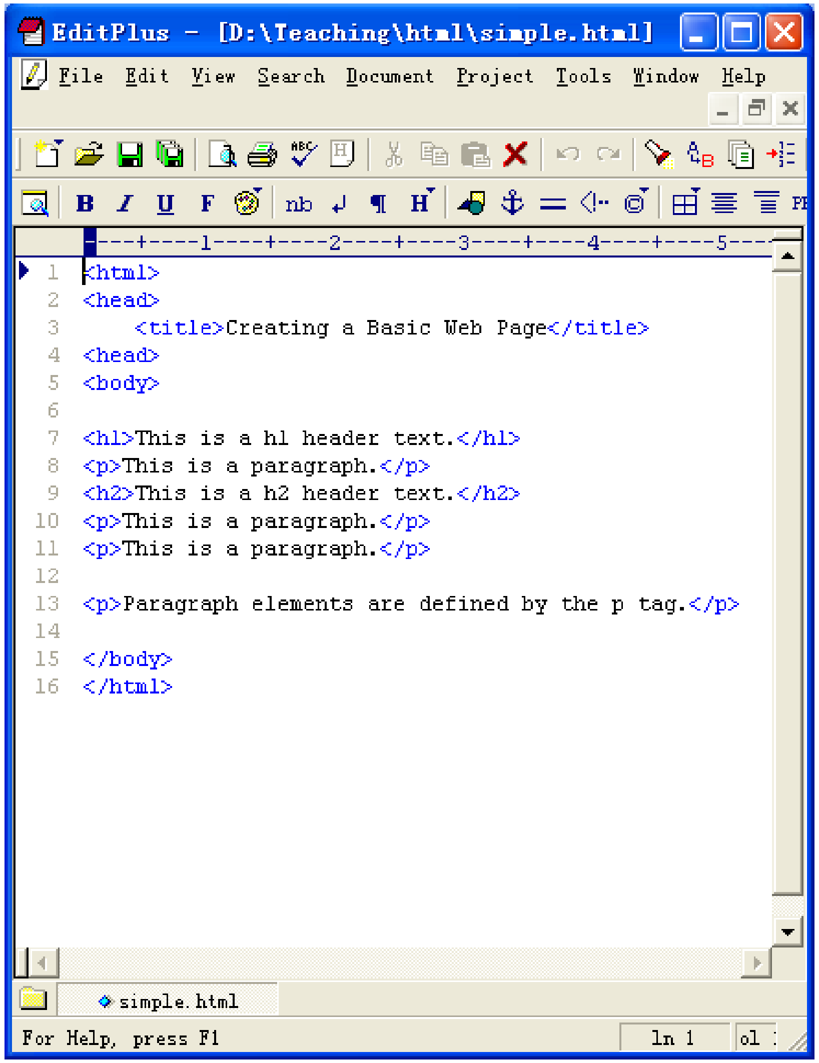
HTML Source Document

- When you connect to a web page by entering its URL into the browser
  - Browser instructs your computer to send a message out over the Internet to the computer specified by that URL requests that it sends back a certain document (**HTML source doc**)
  - **HTML source doc** describes the ***content*** and ***layout*** of the web page
  - After your computer receives the html, your browser interprets the html and displays the resulting web page (text/graphics/links etc)

HTML Source Document

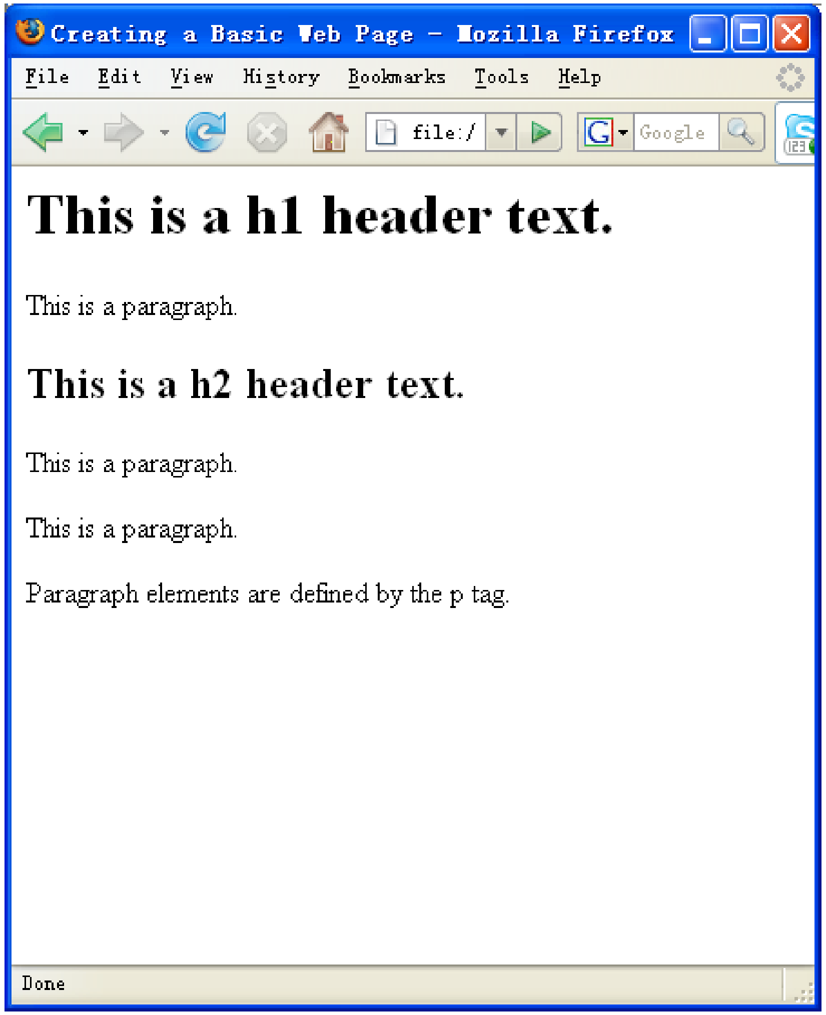
- **HTML** source document
  - A text-only document
  - Consists of (1) actual text, and (2) tags
- A **tag** is an html code that is enclosed in angel brackets <>; used to lay out the web page.
- **XHTML** is a simple, more standardized version of HTML
- XHTML/HTML can be created using a simple text editor like notepad
- File extension must be **.html** or **.htm**

Sample HTML


EditPlus - [D:\Teaching\html\simple.html]
File Edit View Search Document Project Tools Window Help
simple.html
For Help, press F1
ln 1

```
<html>
<head>
    <title>Creating a Basic Web Page</title>
<head>
<body>

<h1>This is a h1 header text.</h1>
<p>This is a paragraph.</p>
<h2>This is a h2 header text.</h2>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>

<p>Paragraph elements are defined by the p tag.</p>

</body>
</html>
```

HTML Source


Creating a Basic Web Page - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:/
Google
Done

This is a h1 header text.

This is a paragraph.

This is a h2 header text.

This is a paragraph.

This is a paragraph.

Paragraph elements are defined by the p tag.

Firefox display of the html source

- **XML** (eXtensible Markup Language):
  - is a **set of rules** that lets web designers classify their data in a way customized to their needs.
  - Extendable by creating new types of tags.
- **XHTML** (eXtensible HyperText Markup Language):
  - A new version of HTML based on XML
  - Inherits strict syntax rules of XML

HTML vs. XHTML

- Some comparisons of HTML vs. XHTML

HTML	XHTML
Tags aren’t extensible	Tags are extensible
Tags are not case-sensitive	Only lowercase tags are allowed
Possible to leave off and ending tag like </body>	Tags should appear in pairs
Overlapping tags	No overlapping tags

- For this course, we use XHTML

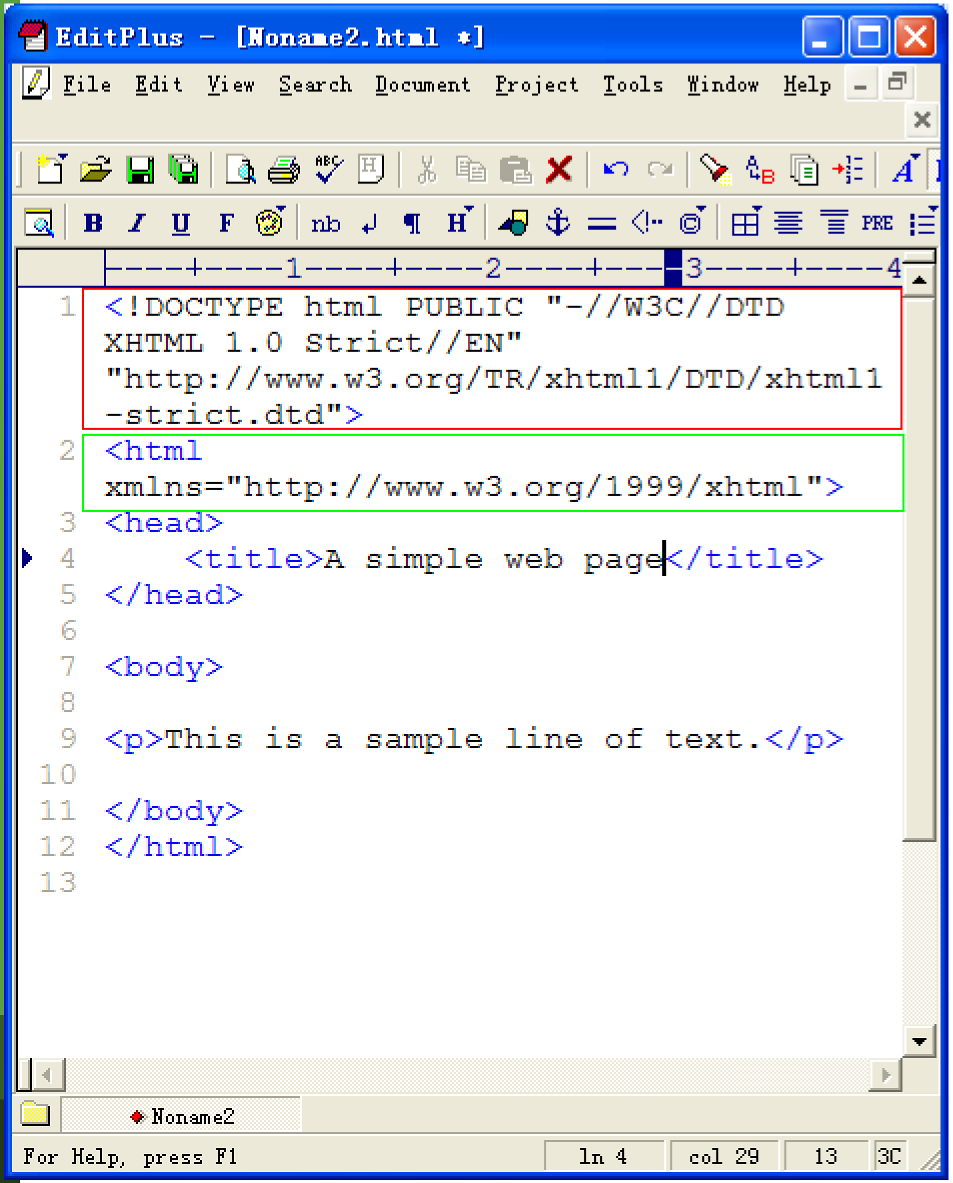
Composition of a XHTML Document

- An XHTML document consists of three main parts:
  - the DOCTYPE
  - the Head
  - the Body

Composition of a XHTML Document

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
        <meta http-equiv="content-type" content="text/html;
charset=utf-8" />
                ...
        <title>…</title>
</head>

<body>
…
</body>
</html>
```

Creating XHTML


EditPlus - [Noname2.html *]
File Edit View Search Document Project Tools Window Help
Noname2
For Help, press F1
ln 4
col 29
13
3C

```
<!DOCTYPE html PUBLIC "-//W3C//DTD
XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1
-strict.dtd">
<html
xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>A simple web page</title>
</head>

<body>

<p>This is a sample line of text.</p>

</body>
</html>
```

The code inside red rectangle (<!DOCTYPE … dtd”>) is a **Document Type Definition (DTD)**, it specifies what type of document this is – in this case an XHTML document.

The code inside green rectangle, **xmlns** specifies the namespace, it tells the browser that all tags contained within the <html> tag belong to the XHTML namespace as defined by the W3C and located at the given URL.

XHTML Tags/Elements

- Tags are also called **elements**
- An **attribute** is a special code that can enhance or modify a tag. They are generally located in the starting tag after the tag name.
- Basic syntax for xhtml tags and attributes
  - **<tag attribute="value">   </tag>**
  - All tags must be lower case
  - all values of attributes need to surrounded by quotes

XHTML Tags/Elements

- Example
  - <strong>This is bold text…</strong>
  - <p style =“text-align:center">This text will appear aligned to the center…</p>

<meta> tag

- **<meta>** tag
  - is used to specify *keywords* that describe a document’s **contents** as well as a **short description**.
- Two necessary attributes – "name" & "content"

```
<meta name="keywords"
  content="baseball, soccer, tennis"/>
<meta name="description"
  content="Sports information page"/>
```

<p> paragraph tag

- **<p>** tag
  - The paragraph tag. Used so separate text within a web page.
  - Container type
  - Will provide line breaks
- Optional attribute : align (not allowed in XHTML 1.0 **Strict** though)

```
<p align="center">
```

<br/> tag

- **<br/>** tag
  - Is used for line break
- Example

```
<p>
 Contact<br />
 6150 Sennott Square<br />
 University of Pittsburgh<br />
 Pittsburgh, PA 15260
 </p>
```

Headings

- **<h1>** to **<h6>**
  - Define headers. <h1> defines the largest header. <h6> defines the smallest header.
- Example

```
<h1>This is header 1</h1>
 <h2>This is header 2</h2>
 <h3>This is header 3</h3>
 <h4>This is header 4</h4>
 <h5>This is header 5</h5>
 <h6>This is header 6</h6>
```

<em> & <strong> tags

- **<em>** tag
  - Renders text as emphasized text
- **<strong>** tag
  - Renders text as strong emphasized text
- Example

```
<em>Emphasized text</em><br />
<strong>Strong text</strong><br />
```

Commenting Source Code

- Comments are inclosed in <!-- and -->
- Example

```
<!--This comment will not be
displayed-->
<p>This is a regular paragraph</p>
```

<blockquote> tag

- **<blockquote>** tag
  - tag defines the start of a long quotation.
- To validate the page as strict XHTML, you must add a block-level element around the text within the <blockquote> tag, like this:

```
<blockquote>
<p>here is a long quotation here is a
long quotation</p>
</blockquote>
```

Attributes, Lists, Tables, Links, and Images

Mehmud Abliz

Block-Level vs. Inline Elements

- This works
  - `<h2><em>Bold and italic</em></h2>`
- How about this
  - `<em><h2>Bold and italic</h2></em>`
- **Block-level element/tag**
  - define a complete section or block of text
  - Can contain inline element and block-level element
- **Inline elements**
  - Define the structure of a sequence of characters within a line
  - may not contain a block-level element
  - may be used within a block

Block-Level vs. Inline Elements

- **Partial list of block-level tags**
  - p, blockquote, h1 … h6, div, ul, ol, li, table, tr, td, th
- **Partial list of inline tags**
  - a (anchor tag), em, strong, img, q (short quotation)

Attribute

- An **attribute** is a special code that can enhance or modify a tag. They are generally located in the starting tag after the tag name.
- Basic syntax for xhtml tags and attributes
  - **<tag attribute="value">   </tag>**
  - All tags must be lower case
  - all values of attributes need to be surrounded by quotes

Common Attributes

- **id**
  - unique identifier for elements
- **class**
  - the class of the element, used to *specify similar attributes for dissimilar elements* by putting them in the same class
- **style**
  - an inline style definition
- **title**
  - a text to display in a tool tip

- **Examples 1**
  - `<p id="firstParag" class="indent" title="This paragraph introduces html attributes">`
  - Assuming style sheet contains
  - `.indent { margin-right: 5%; margin-left: 5%;}`
- **Example 2**
  - `<p id="firstParag" style="margin-right: 5%; margin-left: 5%;" title="This paragraph introduces html attributes">`

Common Attributes

- **lang**
  - sets the language code; “en”: English, “fr”: French, “es”: Spanish, “de”: German etc.
- **dir**
  - sets the text direction, left to right or right to left
- `<p lang=“fr” dir=“ltr”>bonjour!</p>`
- **accesskey**
  - assigns an access key to an element. An **access key** is a single character from the document character set.
- **tabindex**
  - Sets the tab order of an element

Deprecated Attributes

- In order to separate **structure** from **presentation**
  - many HTML attributes/tags used for presentation were deprecated, starting from HTML version 4
- Some deprecated attributes
  - font, `<font size="5" color="red">Text</font>`
  - align, `<p align="center">Centered text</p>`
  - bgcolor, width, height, etc.

Lists

- Ordered lists & Unordered lists
  - `<ol>` for ordered
  - `<ul>` for unordered
  - `<li>` for each item inside the list
- Browser inserts a blank line before & after the list (block-level element)
- Example
  - `<ol> <li>Item 1</li> <li>Item 2</li> <li>Item3</li> </ol>`

Lists

- Nested lists

```
<ul>
  <li>Top Level, Item 1</li>
  <li>Top Level, Item 2
  <ul>
   <li>Sublevel 1, Item 1
   <ul>
         <li>Sublevel 2, Item 1</li>
         <li>Sublevel 2, Item 2</li>
   </ul>
   </li>
   <li>Sublevel 1, Item 2</li>
  </ul>
  </li>
  <li>Top Level, Item 3</li>
</ul>
```

Customizing List Display

- List numbers or marks can be customized
- “**type**” attribute
- Example
  - `<ul type=“square”>`
  - `<ol type=“A”>`
  - `<ol type=“a”>`
  - `<ol type=“I”>`
  - `<ol type=“i”>`
- “**type**” attribute is not allowed in XHTML 1.0 Strict, so use style sheets instead

Definition Lists

- **<dl>** for list element; **<dt>** for “definition terms”; **<dd>** for “definition data”
- Example

  -
    ```
    <dl>
    <dt><strong>CPU</strong></dt>
    <dd>Central Processing Unit</dd>
    <dt><strong>ALU</strong></dt>
    <dd>Arithmetic Logic Unit</dd>
    <dt><strong>GHz</strong></dt>
    <dd>Gigahertz</dd>
    </dl>
    ```

Tables <table>

- Tables used not only for displaying data in tabular format
- A **table** (**<table>**) in HTML
  - Consists of **rows** (**<tr>**)
  - Each row consists of rectangular boxes called **cells** (**<td>**)

  -
    ```
    <table>
    <tr><td>R1,Cell1</td><td>R1,Cell2</td></tr>
    <tr><td>R2,Cell1</td><td>R2,Cell2</td></tr>
    </table>
    ```

Tables

- By default
  - Text in each cell is automatically aligned to the left
  - All the cells in a column have the same width
  - **Width of the column is determined by the cell with the most text in it**
- **<th>** for “**table header**”

```
<tr>
   <th>Header1</th>
   <th>Header2</th>
</tr>
```

Tables

- Other attributes of **<table>**
  - align, cellpadding, cellspacing, colspan
  - Yet XHTML 1.0 Strict don’t allow this attributes, so use stylesheet instead
- Other tags
  - `<caption>`
  - `<colgroup>`
  - `<thead>, <tfoot>, <tbody>`

Links

- The true power of WWW comes with hyperlinks
- Surfer click on a specially marked *word* or *image* on a web page and automatically be jumped to **another web page** or **another place in the same web page**.
  - Another web page – **External link**
  - Another place – **Internal link**
- Use **<a>** (anchor) tag to create a link

Links

- External Links
  - `<a href="`**`SomeURL`**`">Text/image</a>`
- Create a link to CS web page
  - `<a href="http://www.cs.pitt.edu/">CS Department at Pitt</a>`
  - Be careful about the quotation mark
- Internal Links
  Create a place/anchor
  - `<a id="`**`SomeLabel`**`"></a>` or
    `<a id="`**`SomeLabel`**`" name="`**`SomeLabel`**`"/></a>`

  Link to the anchor

  `<a href="#`**`SomeLabel`**`">Go to some place</a>`

Links

- Combining External and Internal Links
  - `<a href=“`**`SomeURL`**`#`**`SomeLabel`**`>Link Text</a>`

Images <img>

- Insert an image using **<img>** tag
  - `<img src="URL of the image file" />`
- Image can an image on a **remote machine** on the Internet, or an image in your **local machine**.
- Examples,
  - `<img src="http://www.cs.pitt.edu/~mehmud/gallery/nature/images/Desert_and_Blue_Sky.jpg"/>`
  - `<img src="../images/Lake_Karakul.jpg" />`

Images

- Alternative Text for images
  - `<img src="Image URL" alt="Alternative Text" />`
- Example
  - `<img src="../images/Lake_Karakul.jpg" alt="Lake Karakul"/>`
- width & height attributes
  - `<img src="../images/Lake_Karakul.jpg" alt="Lake Karakul" width="257" height="161" />`
- Use style sheet instead of attributes, even though XHTML 1.0 Strict supports these two attributes

Cascading Style Sheets (CSS)

Mehmud Abliz

What is CSS?

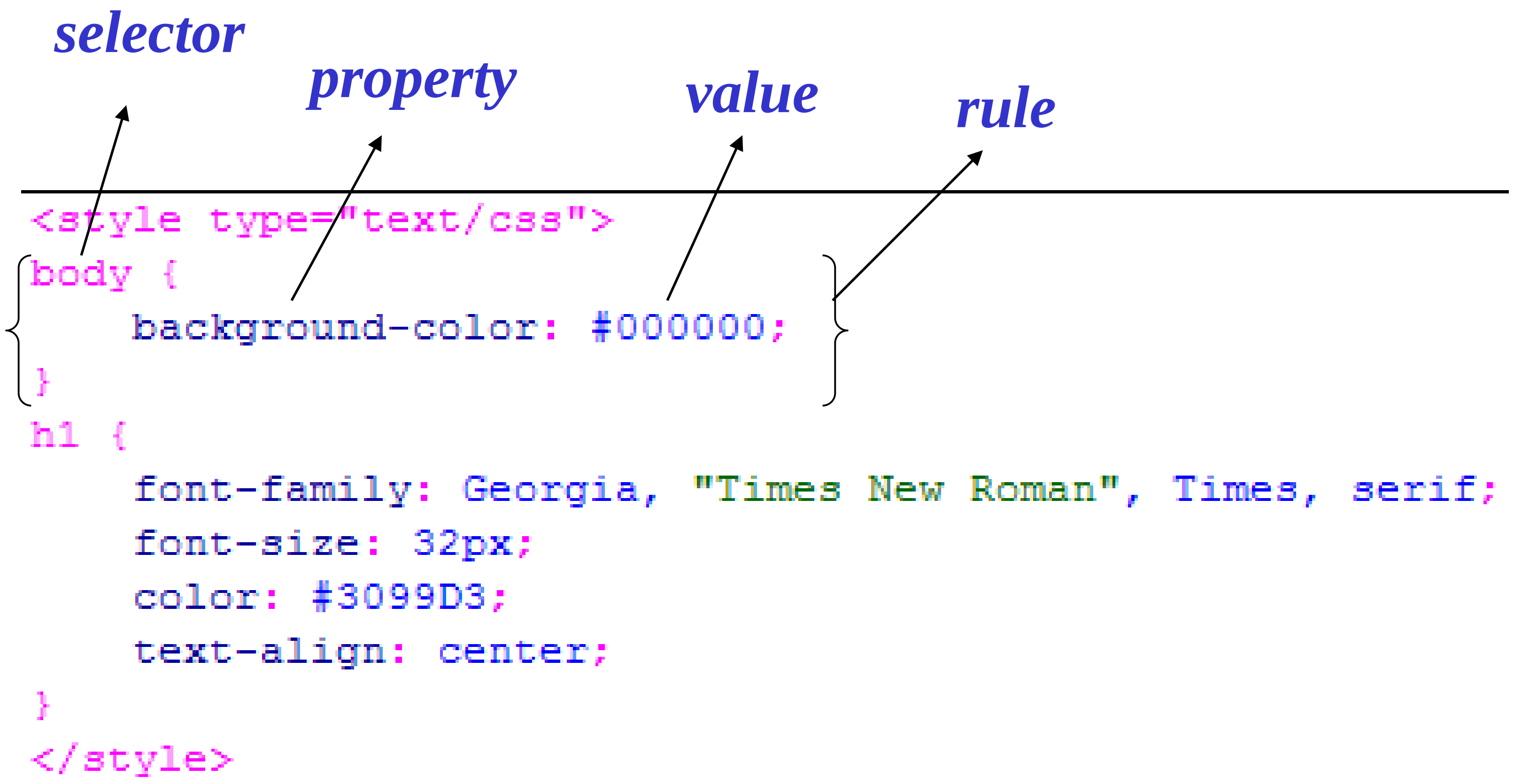
- **Cascading Style Sheets** (**CSS**): is a simple mechanism for *adding style* (e.g. fonts, colors, layouts) to Web documents.
- Styles provide *powerful control* over the **presentation** of web pages.

```
h1 { color: white;
  background: orange;
  border: 1px solid blac
  padding: 0 0 0 0;
  font-weight: bold;
}
/* begin: seaside-theme */

body {
  background-color:white;
  color:black;
  font-family:Arial,sans-serif;
  margin: 0 4px 0 0;
  border: 12px solid;
}
```

Internal Style Sheet

- A style sheet consists of a set of ***rules***.
- Each ***rule*** consists of one or more ***selectors*** and a ***declaration block***.
- A ***declaration block*** consists of a list of ***declarations*** in curly braces ({}).
- Each ***declaration*** consists of a **property**, a **colon** (:), a **value**, then a **semi-colon** (;).

Style Sheet Syntax Explained



```
<style type="text/css">
body {
     background-color: #000000;
}
h1 {
     font-family: Georgia, "Times New Roman", Times, serif;
     font-size: 32px;
     color: #3099D3;
     text-align: center;
}
</style>
```

Basic CSS Syntax

- CSS Syntax
  - **selector {property: value;}**

Three Different Scopes of CSS

- Local
  - confined to a **single element (tag)**
- Internal
  - affect elements in an **entire page**
- External
  - can affect **multiple pages**
- **Precedence**
  - **Local > Internal > External**

Local Style Sheet

- **Example**
  - `<h1 style="color:white; background:orange; font-weight:bold;">Internal Style Sheet Applied to Header 1</h1>`
- **Practice**
  1. add “text-align” property to make it centered
  2. add “border” property to let it has black, 1px thick, solid border at left, right, top, and bottom
- **Tip**: use “border: <top> <right> <bottom> <left>;” format for 4 sides; use “border-<side>: xx yy zz;” for a particular side, where <side> can be left, right, top or bottom. Can apply to other similar properties.

Internal Style Sheet

- How to create?
  - Put **<style> </style>** tag between <head> and </head> tags of your HTML page
  - Use type attribute to indicate the style sheet type, usually type=“text/css”
  - Specify a default style sheet language for the whole document by <meta http-equiv="Content-Style-Type" content="text/css" />
  - Put your set of style sheet ***rules*** in between <style> and </style> tags

Internal Style Sheet

- Practice
  - Create same style in the example in the local style sheet section, but using internal style sheet instead.

External Style Sheet

- An external style sheet is simply a text-only file that contains only CSS **rules**.
- How to link to external style sheet?
  - `<link href="URL of CSS File" rel="stylesheet" type="text/css" />`
- Practice
  - Create a file called “mystyle.css” and do the example in local style sheet, but as external style sheet

Cascading Style Sheets (CSS)

Mehmud Abliz

Selector Type

- **Tag**
  - redefines the look of a specific tag

  E.g. `body {background-color: #000000;}`
- **Class**
  - can apply to any tag

  E.g. `.indent{margin-right:5%;margin-left: 5%;}`

  In HTML, `<p class=“indent”>`
- **Advanced**
  - IDs, pseudo-class selectors

  E.g. `#myId {color: #38608A;}`

Values - Lenghts

- **Lengths** [a number + unit identifier]
  - Unit identifier can be
    em (font size of the relevant font),
    ex (x-height of the relevant font),
    px (pixels),
    in (inches), **cm** (centimeter), **mm**,
    pt (points, =1/72 **in**), **pc** (picas, 1 pc = 12 pt)
  - **E.g.**
    `h1 { margin: 0.5em }, h1 { margin: 1ex }, p { font-size: 12px }, h2 { line-height: 3cm }, h4 { font-size: 12pt }, h4 { font-size: 1pc }`

Values – Percentages & URIs

- **Percentages** [a number + %]
  - `p { line-height: 120% }`
- **URLs & URIs**
  - url("<A URI/URL>"), or
  - url(<A URI/URL>)
  - `li { list-style: url(http://www.cs.pitt.edu/~mehmud/image/bullet2.jpg) disc }`
  - `body { background: url("yellow.png") }`

    where, "yellow" is relative to the URI of the style sheet.

Values - Colors

- **Colors**
  - A color is either a **keyword** (e.g. white, red), or **a numerical RGB specification** (e.g.).
- **A numerical RGB specification** can be:
  - An **RGB value in hexadecimal notation**, which is a ‘#’ immediately followed by a ***6 digit*** or ***3 digit*** **hexadecimal number**, i.e. `#rrggbb` or `#rgb`.
    E.g. `#ff0000, #f00`
  - An **RGB value in functional notation**, i.e. `rgb(rrr,ggg,bbb), rgb(r%,g%,b%)`
    E.g. `rgb(255,0,0), rgb(100%,0%,0%)`

Values - Colors

- 16 original predefined color codes (names)
  - http://www.cs.pitt.edu/~mehmud/cs134/resources/predefined_colors.html
- 216 **browser safe colors**
  - Colors display correctly on all color monitors
  - http://www.cs.pitt.edu/~mehmud/cs134/resources/browser_safe_colors.html

Values - Strings

- **String**
  - Sequence of characters written inside ***double quotes*** ("") or ***single quotes*** ('').
- **Examples**

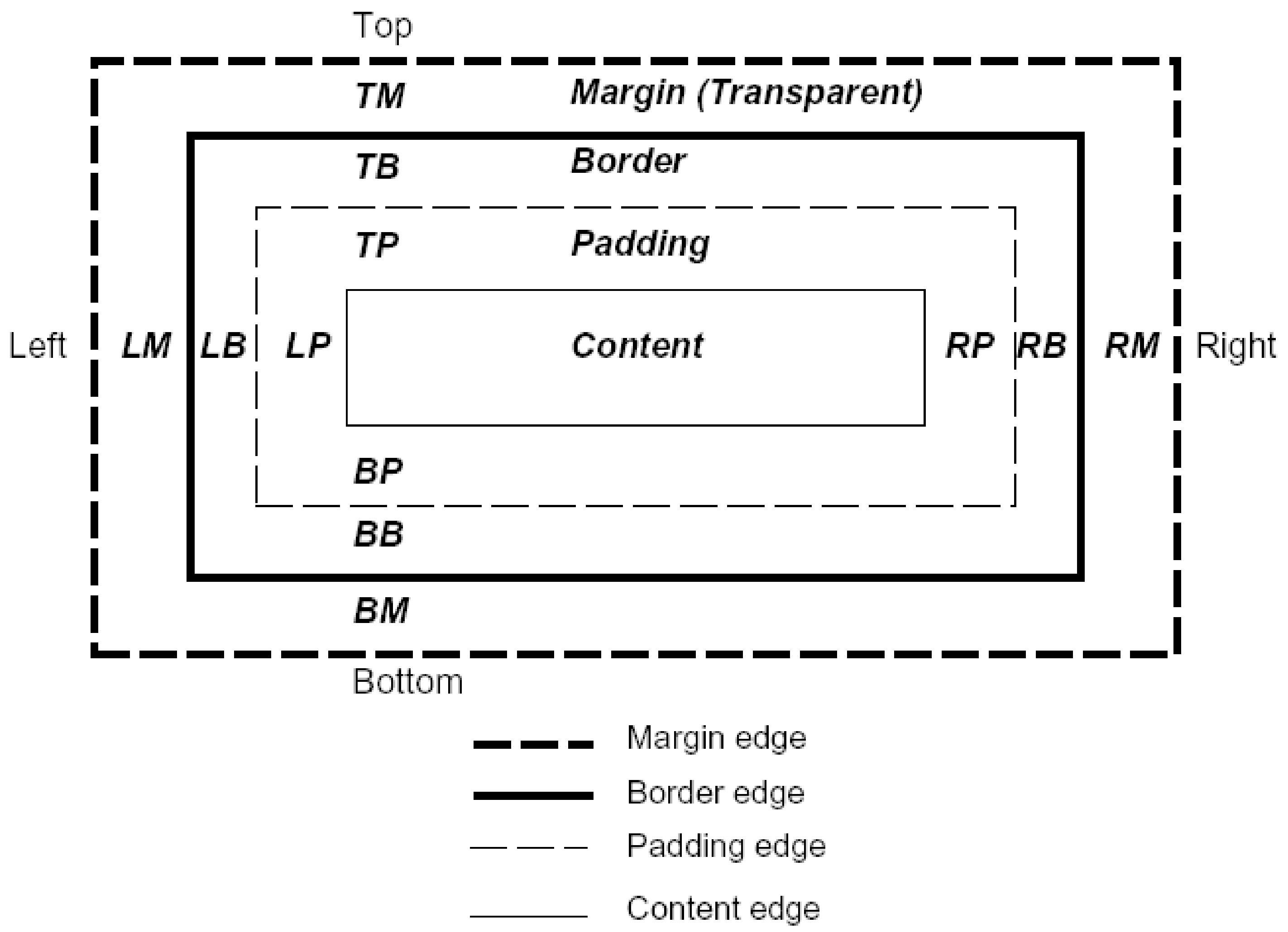
```
"this is a 'string'"
"this is a \"string\""
'this is a "string"'
'this is a \'string\''
```

Box Properties

- margin : <length>
- border : <style> <width> <color>
- padding : <length>
- width & height : <length>
- Examples:

```
p{
        margin: 50px;
        padding: 30px;
        float: right;
        height: 200px;
        width: 400px;
        border: 5px solid #006633;
 }
```

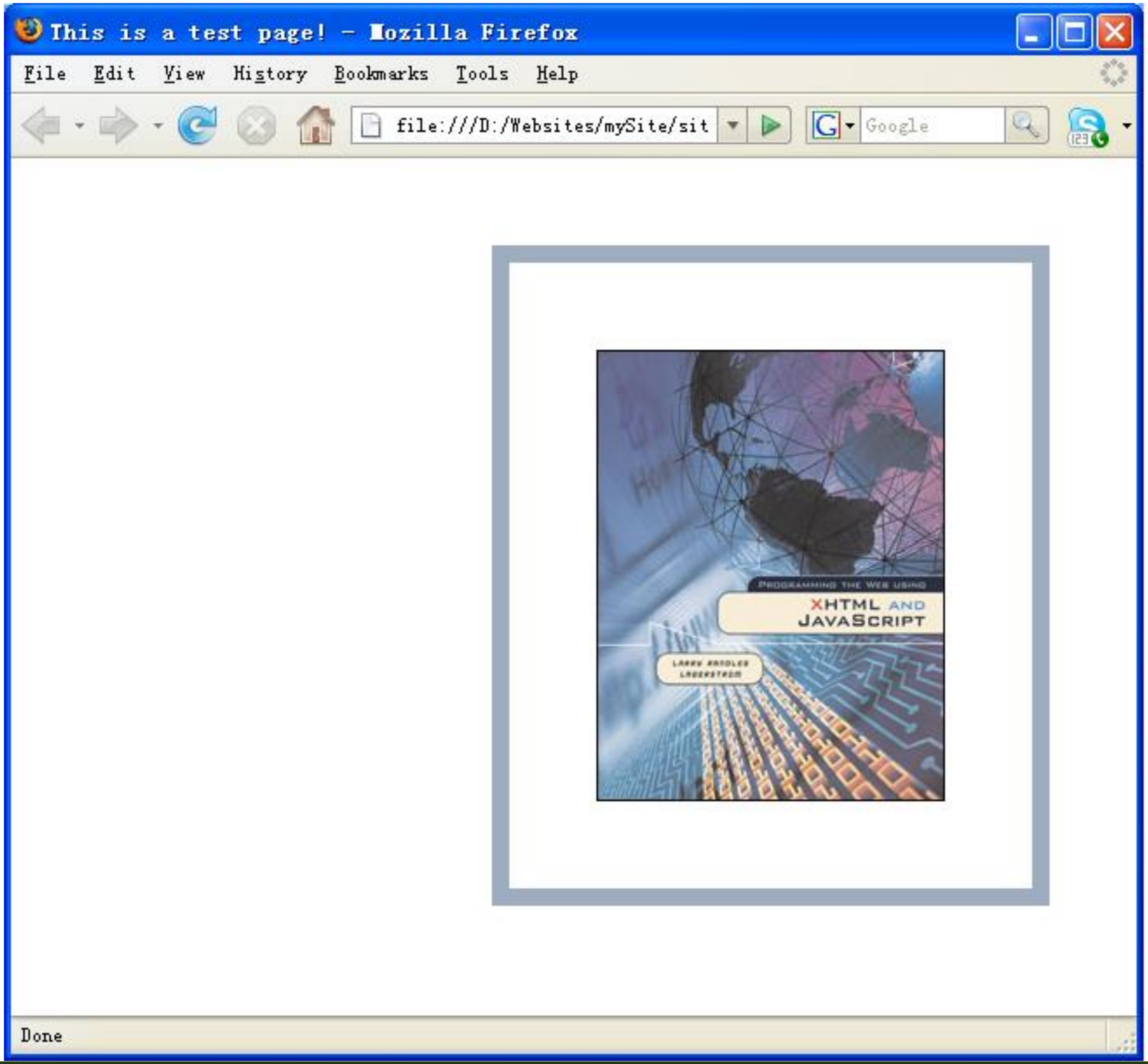
Box Model


Top
TM
Margin (Transparent)
TB
Border
TP
Padding
Left
LM
LB
LP
Content
RP
RB
RM
Right
BP
BB
BM
Bottom
Margin edge
Border edge
Padding edge
Content edge

Box Properties

- Practice
  - Create an internal style called “boxStyle”
  - Download and insert the following image http://www.cs.pitt.edu/~mehmud/cs134/images/lagerstrom_lrg.jpg
  - Apply the “boxStyle” to the image and let it appear as the following image: [in next page]
  - Color value: #9DACBF

Box Property Example


This is a test page! - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///D:/Websites/mySite/sit
Google
Programming the Web using
XHTML and JavaScript
Larry Randles Lagerstrom
Done

Text Properties

- **font-family** : <font name>, | <font name>, …
- **font-size** : <length> | <percentage> | inherit
- **font-weight** : normal | bold | lighter | 100 | 200 ...
  - normal = 400, bold = 700, lighter is relative
- **font-style** : normal | italic | oblique | inherit
- **line-height** : normal | <length> | <percentage> | inherit
- **text-transform** : capitalize | uppercase | lowercase | none | inherit
- **color** : <color>
- **text-indent** : <length> | <percentage> | inherit
- **text-align** : left | right | center | justify | inherit

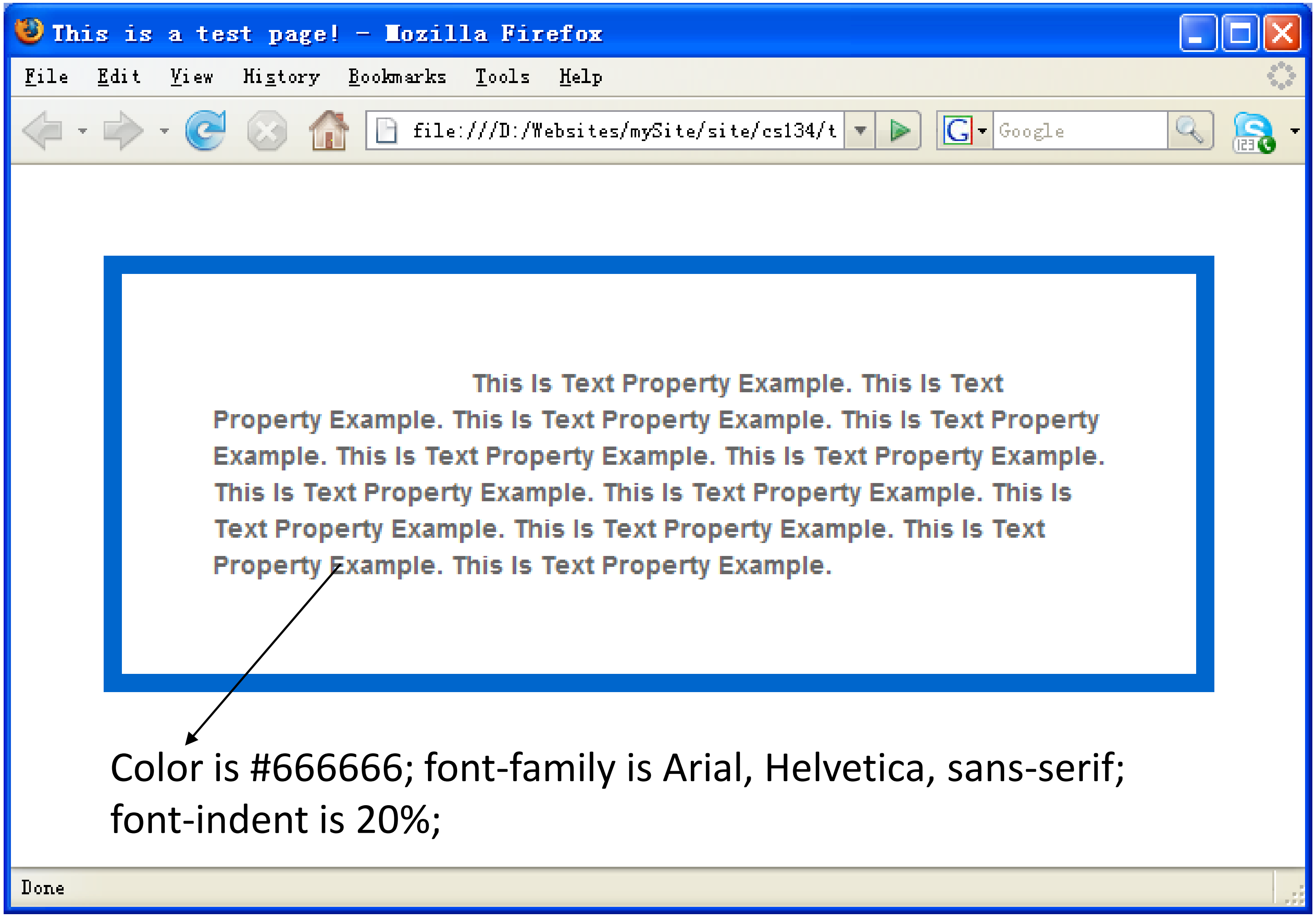
Text Properties Practice

- Practice
  - Create a paragraph text
  - Create internal style for <p> tag as following

```
p{
        margin: 50px;
        padding: 50px;
        clear: right;
        float: right;
        border: 10px solid #0066CC;
}
```

  - Create a internal style called “textStyle” and apply it to paragraph text and let it look like this

Text Properties Practice


This is a test page! - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///D:/Websites/mySite/site/cs134/t
Google
Done

This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example. This Is Text Property Example.

Color is #666666; font-family is Arial, Helvetica, sans-serif; font-indent is 20%;

Positioning Properties

- **height** : <length> | <percentage> | inherit
- **width** : <length> | <percentage> | inherit
- **left** : <length> | <percentage> | auto | inherit
- **top** : <length> | <percentage> | auto | inherit
- **right** : <length> | <percentage> | auto | inherit
- **bottom** : <length> | <percentage> | auto | inherit
- **position** : static | relative | absolute | fixed | inherit
- **left/top/right/bottom** usually used when position is specified as **absolute**.

Values for position property

Value	Description
static	Default. An element with position: static always has the position the normal flow of the page gives it (a static element ignores any top, bottom, left, or right declarations)
relative	An element with position: relative moves an element relative to its normal position, so "left:20" adds 20 pixels to the element's LEFT position
absolute	An element with position: absolute is positioned at the specified coordinates relative to its containing block. The element's position is specified with the "left", "top", "right", and "bottom" properties
fixed	An element with position: fixed is positioned at the specified coordinates relative to the browser window. The element's position is specified with the "left", "top", "right", and "bottom" properties. The element remains at that position regardless of scrolling. Works in IE7 (strict mode)

Positioning Properties

- A floated box is shifted to the left or right until its outer edge touches the ***containing block*** *edge* or the *outer edge of another float*.
- Example

```
.positionStyle {
        height: 400px;
        width: 200px;
        left: 50px;
        top: 50px;
        right: 50px;
        bottom: 50px;
        position:absolute;
        float: rigth;
}
```

Positioning Properties

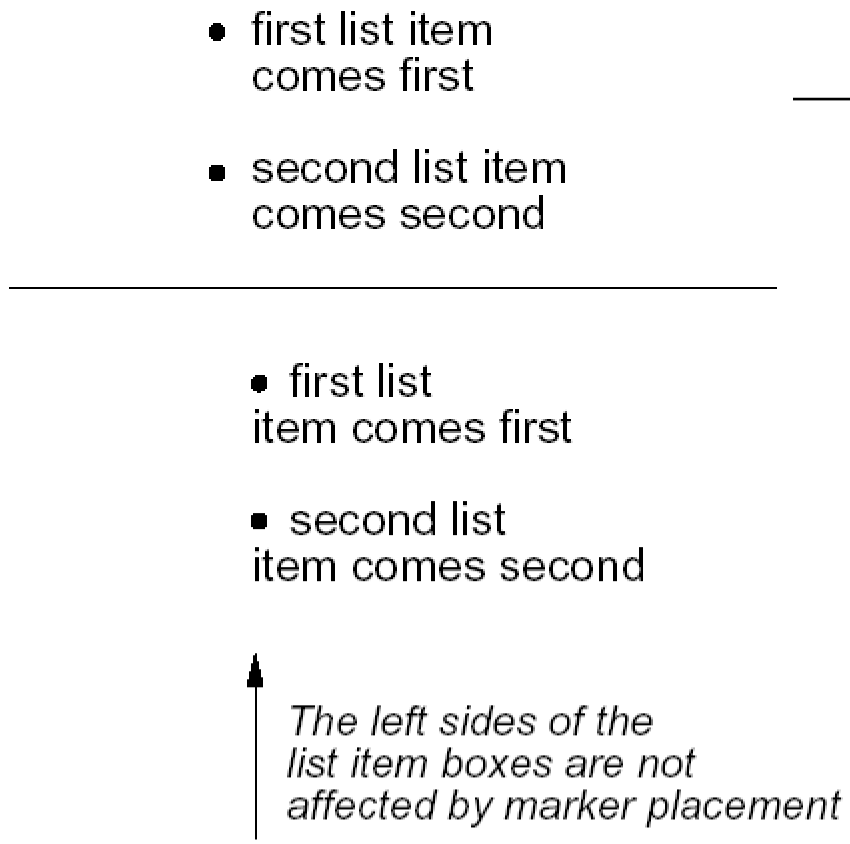
- Practice – thumbnail image
  - Create a table with only one cell
  - Insert the following image 10 times inside the only cell of the table http://www.cs.pitt.edu/~mehmud/image/Muztagh_Ata.jpg
  - Create a rule called ".thumbImg" which floats to left, has a margin of 20 pixels on the top & bottom and 15 pixels on the left & right; has a 5-pixel solid border with silver color
  - Apply the rule to all the 10 images

List Properties

- **list-style**: [disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha | upper-alpha | none] || [inside | outside] || [<url> | none]
  - **Example**
    ```
    li { list-style:
    url(http://www.cs.pitt.edu/~mehmud/image/bu
    llet2.jpg) disc inside; }
    ```
- Can use the **list-style-type**, **list-style-image**, **list-style-position** separately.

List Properties

Difference between *inside* and *outside*



$\longrightarrow$ Outside

Box Model for List

```
 _______________________________________________________
|                                                       |
|    A      UL margin (transparent)                     |
|    _______________________________________________    |
| D |                                               | B |
|   |    E   UL padding (red)                       |   |
|   |    _______________________________________    |   |
|   | H |                                       | F |   |
|   |   |    a   LI margin (transparent,        |   |   |
|   |   |        so red shines through)         |   |   |
|   |   |    _______________________________    |   |   |
|   |   | d |                               | b |   |   |
|   |   |   |    e   LI padding (blue)      |   |   |   |
|   |   |   |                               |   |   |   |
|   |   |   | h  1st element of list      f |   |   |   |
|   |   |   |                               |   |   |   |
|   |   |   |    g                          |   |   |   |
|   |   |   |_______________________________|   |   |   |
|   |   |                                       |   |   |
|   |   |    max(a, c)                          |   |   |   <- note the max
|   |   |    _______________________________    |   |   |
|   |   |   |                               |   |   |   |
|   |   | d |    e   LI padding (blue)      |   |   |   |
|   |   |   |                               |   |   |   |
|   |   |   | h  2nd element of list      f |   |   |   |
|   |   |   |                               |   |   |   |
|   |   |   |    g                          |   |   |   |
|   |   |   |_______________________________|   |   |   |
|   |   |                                       |   |   |
|   |   |   c    LI margin (transparent,        |   |   |
|   |   |        so red shines through)         |   |   |
|   |   |_______________________________________|   |   |
|   |                                               |   |
|   |    G                                          |   |
|   |_______________________________________________|   |
|                                                       |
|   C                                                   |
|_______________________________________________________|
```

Background Properties

- **background-color**: <color> | transparent | inherit
- **background-image**: <uri> | none | inherit
- **background-position**: [[<percentage> | <length> | left | center | right] [<percentage> | <length> | top | center | bottom]?] | [[left | center | right] || [top | center | bottom]] | inherit
- **background-repeat**: repeat | repeat-x | repeat-y | no-repeat | inherit
- **background-attachment**: scroll | fixed | inherit

Background Properties

- Practice

```
body {
        background-image:
        url(http://www.cs.pitt.edu/~mehmud/imag
e/bgd.png);
        background-repeat:no-repeat;
        color: #FFFFFF;
        background-position:center center;
        background-color: #666666;
        background-attachment: fixed;
}
```

<div> and <span> tags

- **<div>**
  - is a generic block-level tag
- **<span>**
  - is a generic inline tag, it **spans** a series of characters
- **Example**

```
.subTitle {
        font-family: Georgia, Times, serif;
        font-size: 14px; font-weight: bold; color: #960000;
}
```

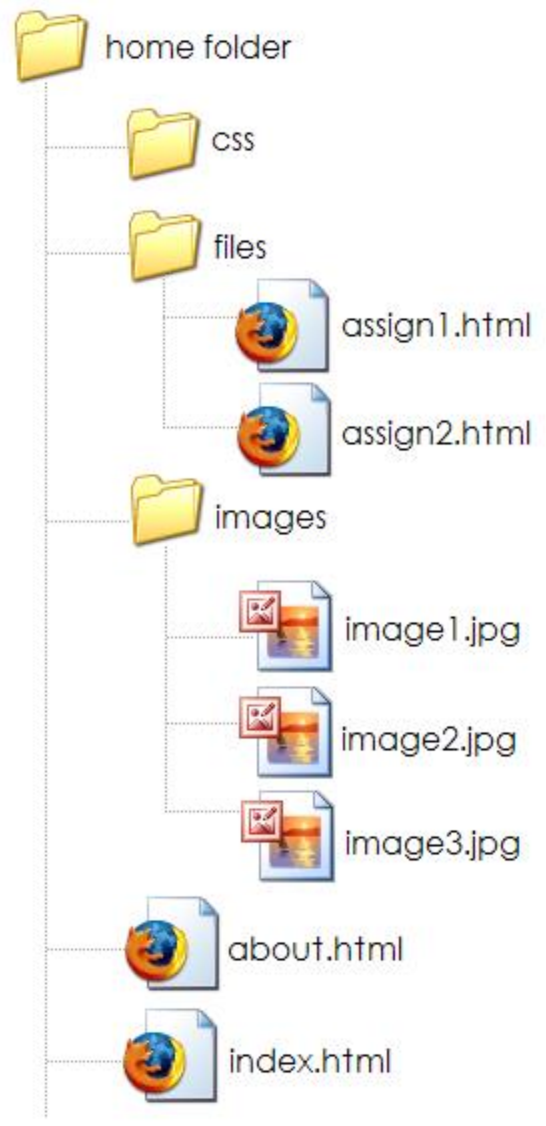
```
<p>blah blah blah <span class="subTitle">some text in different
style</span> text text text text</p>
```

<title> tag

- **<title>**:
  - defines the title of the document .
  - Placed between **<head>** and **</head>**
- **Example**

```
<html>

<head>
<title>XHTML Tag Reference</title>

…
</head>

…

</html>
```

Directory Structure



Understanding URL & Learn How to Use Different CSS Selector Type

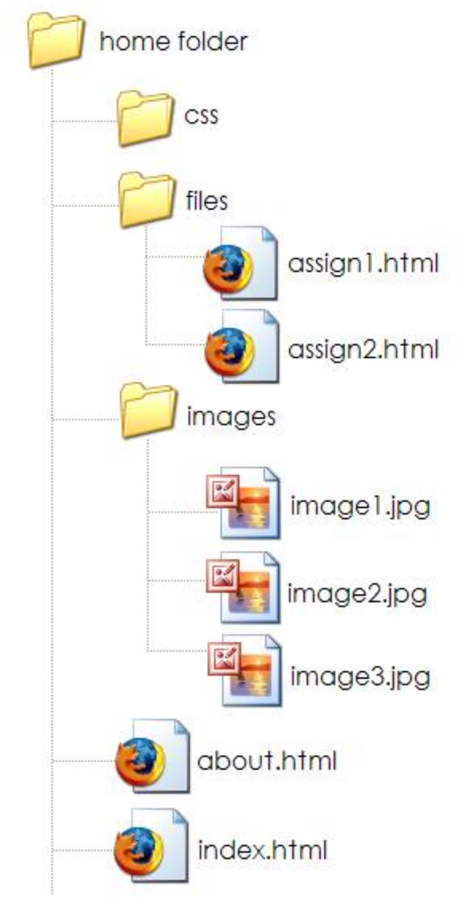
Mehmud Abliz

Understanding URL

- When we have html code like following:
  <img src=“SOME URL“ />
  <a href=“SOME URL"> </a>
- Or CSS code like:
  background: url(../img/pdf.png) no-repeat;
  list-style: url(./img/bullet2.png) inside;
- We need to specify the URL.
- How do we know the URL of an object (a file, or a link)?

Understanding URL

- Assume we have following folders and files



Understanding URL

- If we want to insert "image1.jpg“ in the "assign2.html" file, then the html code will be:
  `<img src="../images/image1.jpg" />`

Understanding URL

- **Practice**:
  - download the "c8_e1.zip“ file from the following address: http://www.cs.pitt.edu/~mehmud/cs134-2084/files/c8_e1.zip
  - 1) Insert "image2.jpg" in "about.html" file;
    2) Insert "image3.jpg" in "index.html" and also in "assign1.html";
    3) Make a link to "about.html", "assign1.html", "assign2.html" files in the "index.html" file.

Using Different CSS Selectors

- **Type Selector**
  - redefines the look of a specific tag
  - A type selector matches every instance of the element type in the document tree

E.g. `body {background-color: #000000;}`

- **Class Selector**
  - can apply to any tag

E.g. `.indent{margin-right:5%;margin-left: 5%;}`

In HTML, `<p class="indent">`

- **ID Selector**

E.g. `#myId {color: #38608A;}`

In HTML, `<h1 id="myId">some text..</h1>`

Using Different CSS Selectors

- **Grouping**
  - When several selectors share the same declarations, they may be grouped into a comma-separated list.

    e.g. `h1, h2, h3 {font-family: Georgia;}`
- **Universal selector**
  - The universal selector, written "`*`", matches the name of any element type.
  - **e.g.** `* {border: 2px;}`

- **Descendant selector**
  - Sometimes, you want selectors to match *an element that is the descendant of another element* in the document tree (**e.g.**, "Match those EM elements that are contained by an H1 element").
  - Descendant selectors express such a relationship in a pattern.
  - A descendant selector is made up of **two or more selectors separated by whitespace**.

e.g. `h1 em {color: blue;}`

- **When to use which?**
  - Use "**type selector**" when you want to apply certain style for all occurrences of a certain tag
  - Use "**ID selector**" if you want to apply the style for only one occurrence of a certain tag
  - Use "**class selector**" if you want to apply the style for many (but not all) occurrences of a certain tag; *OR* if you want to apply the style for more than one type of tags

- **When to use which? (cont.)**
  - Use “**Grouping**” When several selectors share the same declarations
  - Use “**Universal selector**” if you want all the tags in your web documents have some common style (for example, all tags don’t have any margin)
  - Use “**Descendant selectors**” if you want selectors to match an element that is the descendant of another element

Using Different CSS Selectors

- **Practice**
  - Save the HTML file at the following address to your local hard drive: **http://www.cs.pitt.edu/~mehmud/cs134-2084/exercise/orion.html**
  - Add following CSS rules into orion.html as internal stylesheet:
    *, h1, h2, h6,
     h1, h2, h6
    , p, .yellowBG, #topcontent, ul li a, #selectedli
  - I explain how you should add them

Web Page Layout

Mehmud Abliz

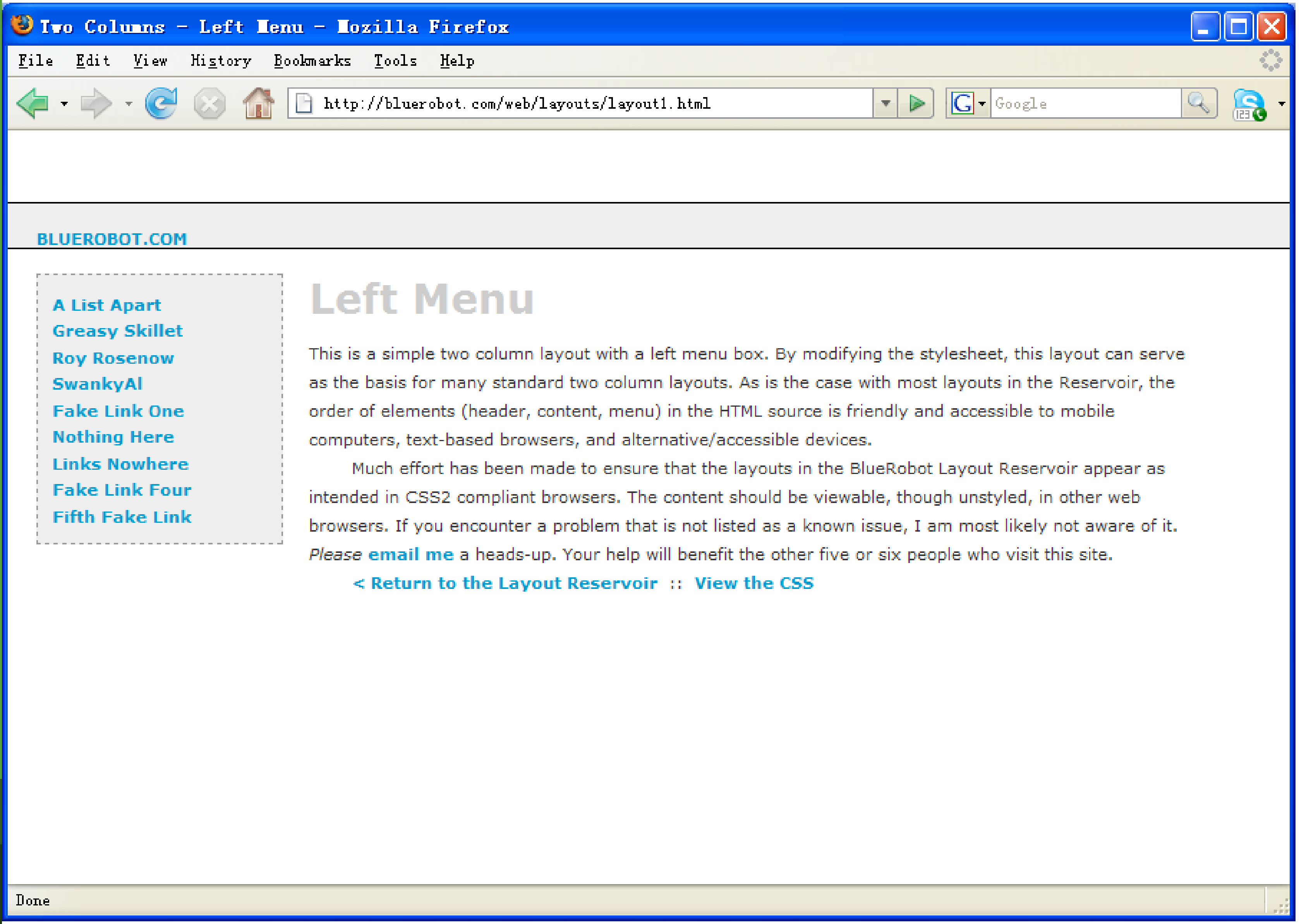
table vs. div

- **table**
  - Pros: supported by all browsers
  - Cons: bind style to content; hard to maintain
- **div**
  - Pros: easy to maintain
  - Cons: not supported by all browsers
- We recommend – **div**, reasons:
  - CSS is to separate **structure** from **content**.
  - Supporting most common/popular browsers are enough. May be it’s time for some people to upgrade their browsers.

Common Layouts

- Variable width content:
  - 2 columns - left menu
  - 2 columns - right menu
  - 3 columns
- Centered (fixed width content):
  - 2 columns
  - 3 columns
- 4 columns (fluid/variable width)
- Many other …

2 columns - left menu


Two Columns - Left Menu - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://bluerobot.com/web/layouts/layout1.html
Google
Done

BLUEROBOT.COM

- A List Apart
- Greasy Skillet
- Roy Rosenow
- SwankyAl
- Fake Link One
- Nothing Here
- Links Nowhere
- Fake Link Four
- Fifth Fake Link

Left Menu

This is a simple two column layout with a left menu box. By modifying the stylesheet, this layout can serve as the basis for many standard two column layouts. As is the case with most layouts in the Reservoir, the order of elements (header, content, menu) in the HTML source is friendly and accessible to mobile computers, text-based browsers, and alternative/accessible devices.

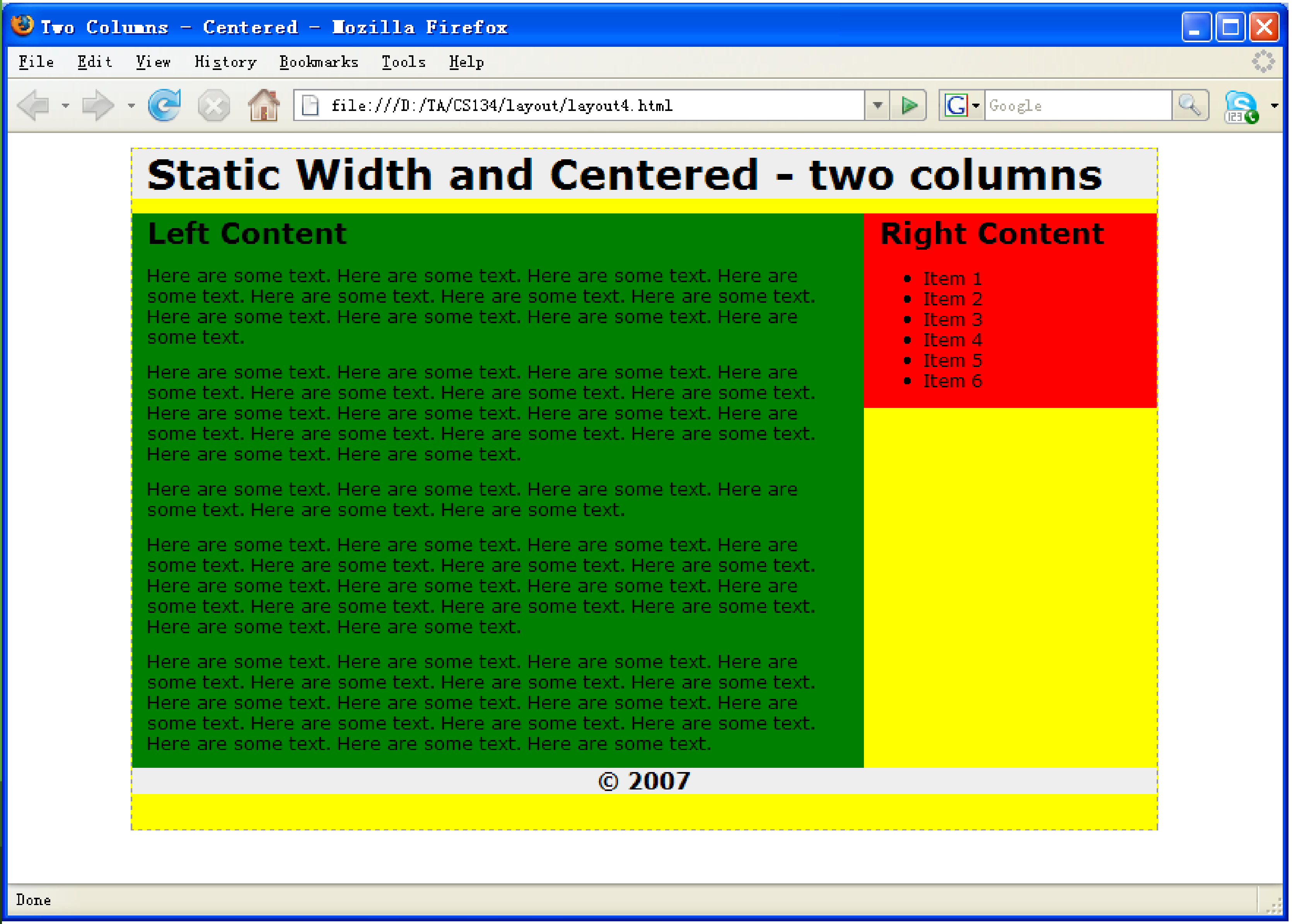
Much effort has been made to ensure that the layouts in the BlueRobot Layout Reservoir appear as intended in CSS2 compliant browsers. The content should be viewable, though unstyled, in other web browsers. If you encounter a problem that is not listed as a known issue, I am most likely not aware of it. *Please* **email me** a heads-up. Your help will benefit the other five or six people who visit this site.

< Return to the Layout Reservoir :: **View the CSS**

2 columns - left menu

```
#Header {
       margin:50px 0px 10px 0px;
       padding:17px 0px 0px 20px;
       border:1px dashed #999;
       background-color:#eee;
}
#Content {
       margin:0px 50px 50px 200px;
       padding:10px;
       border:1px dashed #999;
       background-color: #eee;
}
#Menu {
       position:absolute;
       top:100px;
       left:20px;
       width:150px;
       padding:10px;
       background-color:#eee;
       border:1px dashed #999;
}
```

2 columns – centered, static width


Two Columns - Centered - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///D:/TA/CS134/layout/layout4.html
Google
Done

Static Width and Centered - two columns

Left Content

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Right Content

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6

© 2007

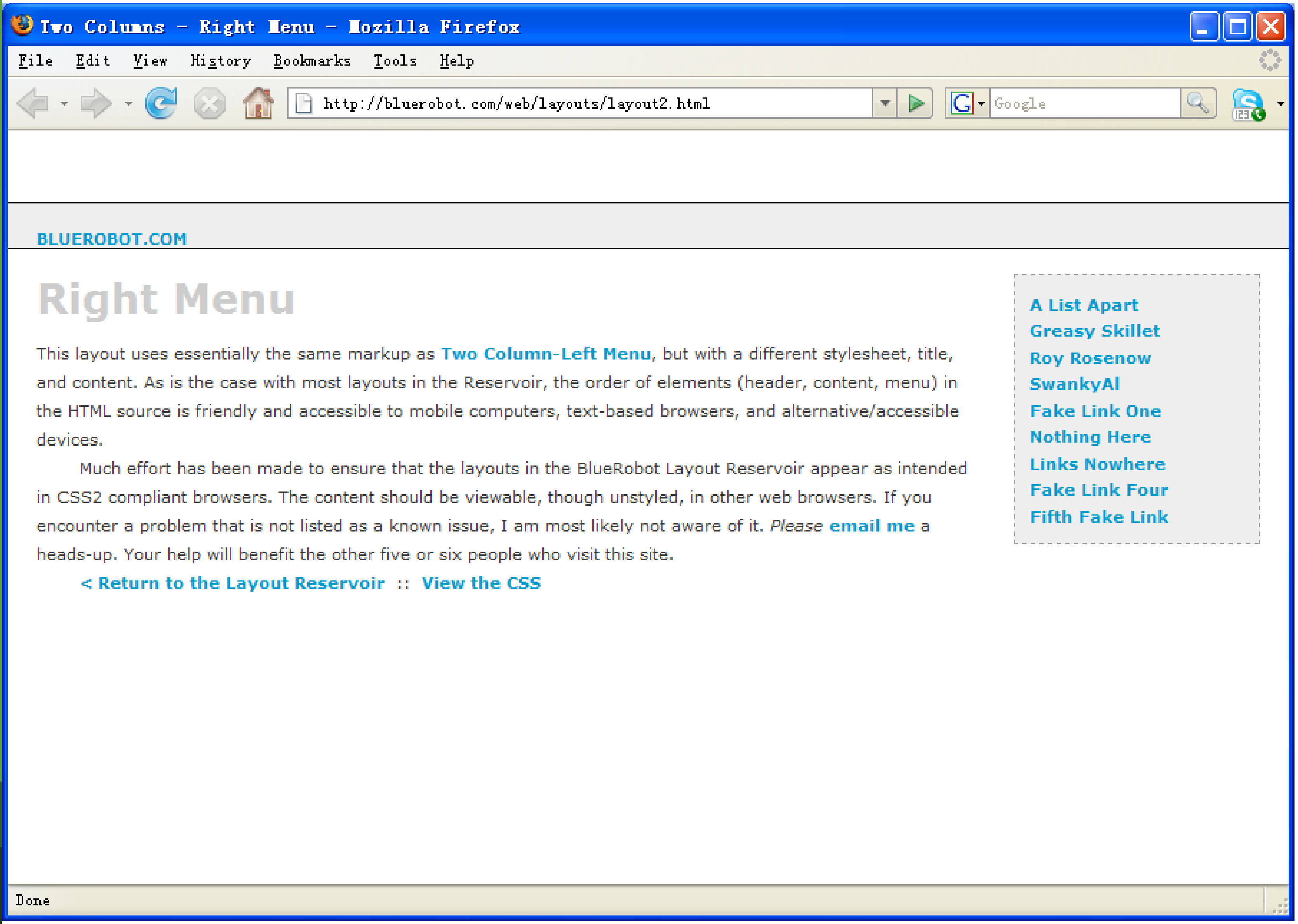
2 columns – centered, static width

```
body {
	margin:0px;
	padding:0px;
	text-align: center;
}

#Wrapper {
	width:700px;
	margin-right:auto;
	margin-left:auto;
	border:1px dashed #999;
}

#Header {
	background: #eee;
}
```

2 columns – centered, static width

```
#Menu {
      float:right;
      width:200px;
      background: #eee;
}

#Content {
      float:left;
      width:500px;
      background: #666;
}

#Footer {
      clear: both;
      background: #eee;
}
```

2 columns - right menu


Two Columns - Right Menu - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://bluerobot.com/web/layouts/layout2.html
Google
Done

BLUEROBOT.COM

Right Menu

This layout uses essentially the same markup as **Two Column-Left Menu**, but with a different stylesheet, title, and content. As is the case with most layouts in the Reservoir, the order of elements (header, content, menu) in the HTML source is friendly and accessible to mobile computers, text-based browsers, and alternative/accessible devices.

Much effort has been made to ensure that the layouts in the BlueRobot Layout Reservoir appear as intended in CSS2 compliant browsers. The content should be viewable, though unstyled, in other web browsers. If you encounter a problem that is not listed as a known issue, I am most likely not aware of it. *Please* **email me** a heads-up. Your help will benefit the other five or six people who visit this site.

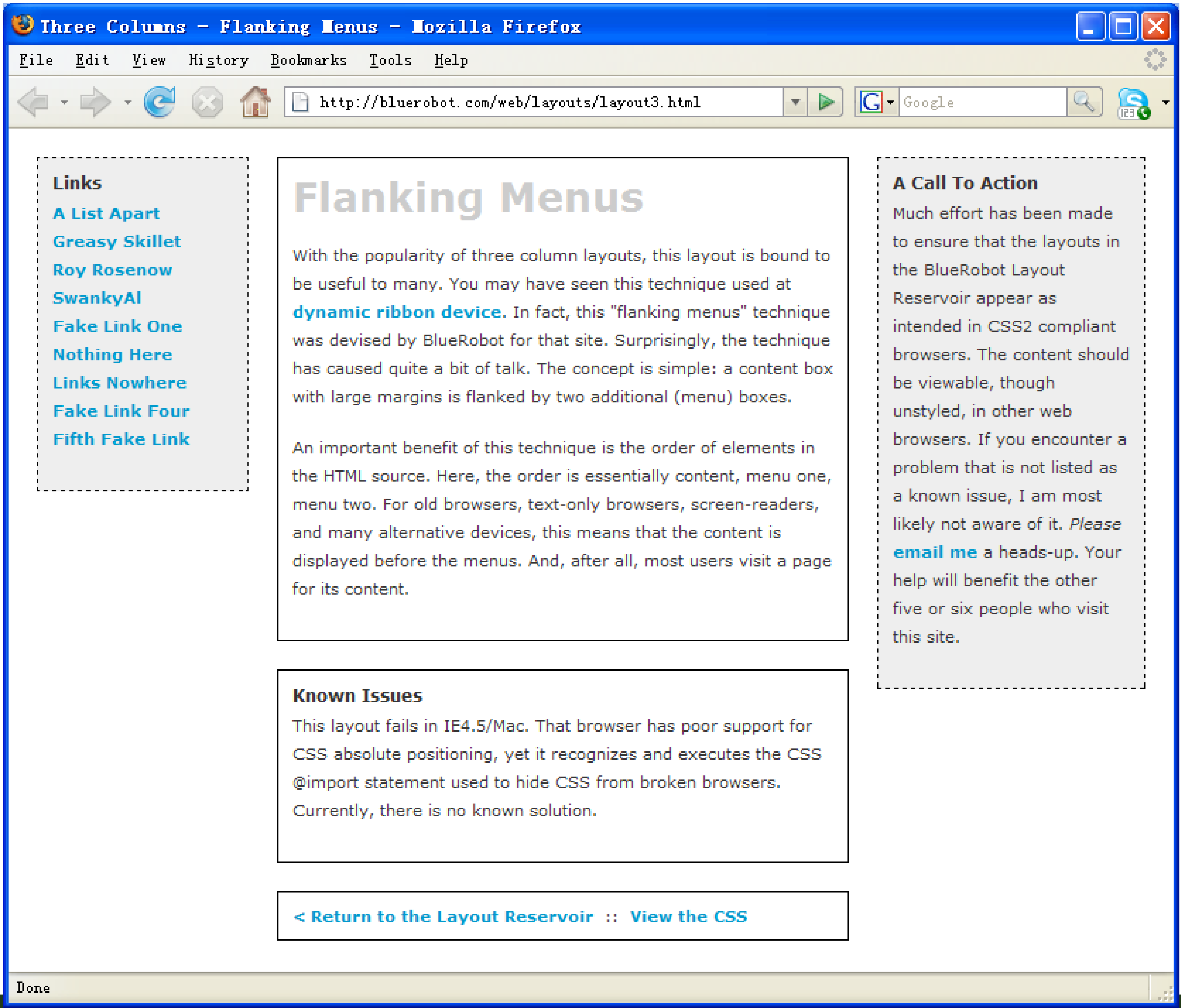
< Return to the Layout Reservoir :: **View the CSS**

A List Apart
Greasy Skillet
Roy Rosenow
SwankyAl
Fake Link One
Nothing Here
Links Nowhere
Fake Link Four
Fifth Fake Link

2 columns - right menu

```
#Header {
       margin:50px 0px 10px 0px;
       padding:17px 0px 0px 20px;
       border:1px dashed #999;
       background-color:#eee;
}
#Content {
       margin:0px 200px 50px 50px;
       padding:10px;
       border:1px dashed #999;
       background-color: #eee;
}
#Menu {
       position:absolute;
       top:100px;
       right:20px;
       width:150px;
       padding:10px;
       background-color:#eee;
       border:1px dashed #999;
}
```

3 columns - flanking menu


Three Columns - Flanking Menus - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://bluerobot.com/web/layouts/layout3.html
Google
Done

Links

- A List Apart
- Greasy Skillet
- Roy Rosenow
- SwankyAl
- Fake Link One
- Nothing Here
- Links Nowhere
- Fake Link Four
- Fifth Fake Link

Flanking Menus

With the popularity of three column layouts, this layout is bound to be useful to many. You may have seen this technique used at **dynamic ribbon device.** In fact, this "flanking menus" technique was devised by BlueRobot for that site. Surprisingly, the technique has caused quite a bit of talk. The concept is simple: a content box with large margins is flanked by two additional (menu) boxes.

An important benefit of this technique is the order of elements in the HTML source. Here, the order is essentially content, menu one, menu two. For old browsers, text-only browsers, screen-readers, and many alternative devices, this means that the content is displayed before the menus. And, after all, most users visit a page for its content.

Known Issues

This layout fails in IE4.5/Mac. That browser has poor support for CSS absolute positioning, yet it recognizes and executes the CSS @import statement used to hide CSS from broken browsers. Currently, there is no known solution.

< Return to the Layout Reservoir :: View the CSS

A Call To Action

Much effort has been made to ensure that the layouts in the BlueRobot Layout Reservoir appear as intended in CSS2 compliant browsers. The content should be viewable, though unstyled, in other web browsers. If you encounter a problem that is not listed as a known issue, I am most likely not aware of it. *Please* **email me** a heads-up. Your help will benefit the other five or six people who visit this site.

3 columns - flanking menu

```
.content {
	position:relative;
	width:auto;
	min-width:120px;
	margin:0px 210px 20px 170px;
	border:1px solid black;
	padding:10px;
	z-index:3; /* This allows the content to overlap the right menu in narrow windows in good browsers. */
}
#navAlpha {
	position:absolute;
	width:128px;
	top:20px;
	left:20px;
	border:1px dashed black;
	background-color:#eee;
	padding:10px;
	z-index:2;
}
```

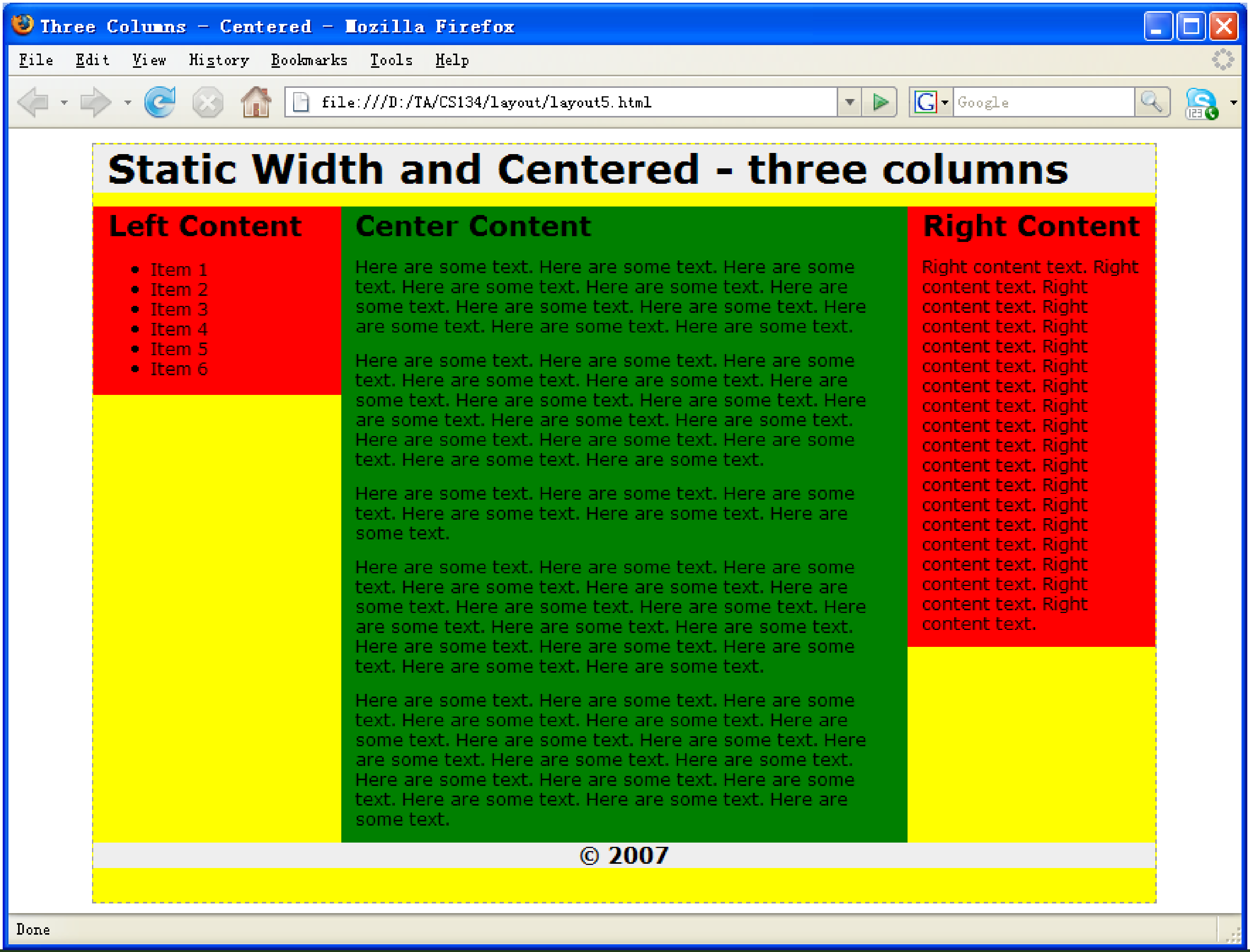
3 columns - flanking menu

```
#navBeta {
        position:absolute;
        width:168px;
        top:20px;
        right:20px;
        border:1px dashed black;
        background-color:#eee;
        padding:10px;
        z-index:1;
}
```

z-index (or stack level)

- **z-index**: auto | <integer> | inherit
  - Z-axis positions are particularly relevant when boxes overlap visually.
  - In addition to their horizontal and vertical positions, boxes lie along a "z-axis" and are formatted one on top of the other.
  - Boxes with higher z-index stacked on top of the boxes with lower z-index.
  - Boxes with the z-index are stacked back-to-front according to document tree order.

3 columns – centered, static width


Three Columns - Centered - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///D:/TA/CS134/layout/layout5.html
Google
Done

Static Width and Centered - three columns

Left Content

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6

Center Content

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Right Content

Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text.

© 2007

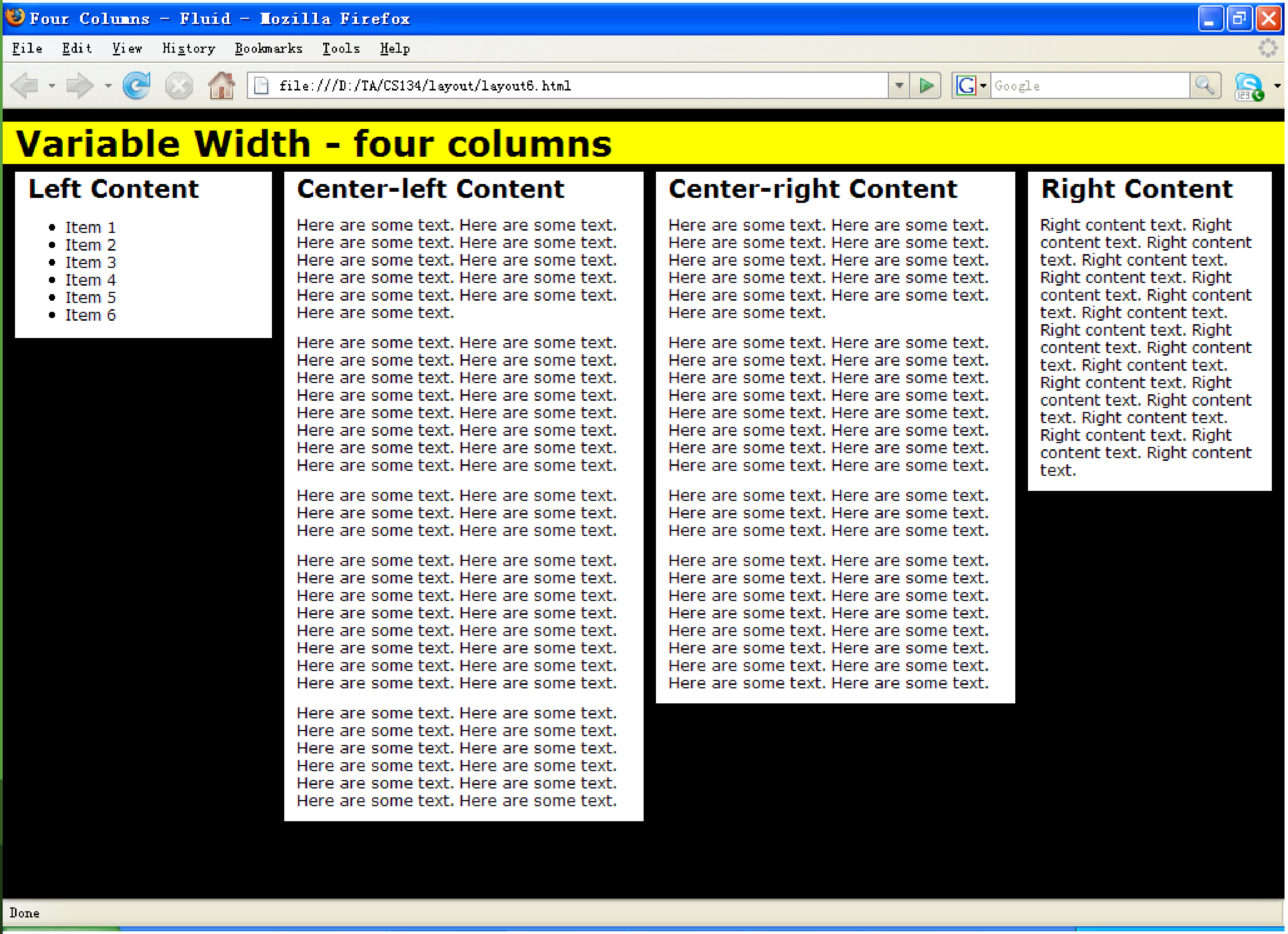
3 columns – centered, static width

```css
body {
	text-align:center;
	margin:0px;
	padding:0px;
	font:12px verdana, arial, helvetica, sans-serif;
}

#frame {
	width:750px;
	margin-right:auto;
	margin-left:auto;
	margin-top:10px;
	text-align:left;
	border:1px dashed #999;
	background-color: yellow;
}

#topcontent {
	background-color: #eee;
}
```

3 columns – centered, static width

```
#centercontent {
        float:left;
        width:400px;
        background-color: green;
}
#leftcontent {
        float:left;
        width:175px;
        background-color: red;
}
#rightcontent {
        float:left;
        width:175px;
        background-color: red;
}

#bottomcontent {
        background-color:#eee;
        text-align:center;
}
```

4 columns - variable width


Four Columns - Fluid - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///D:/TA/CS134/layout/layout6.html
Google
Done

Variable Width - four columns

Left Content

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5
- Item 6

Center-left Content

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Center-right Content

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text. Here are some text.

Right Content

Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text. Right content text.

4 columns - variable width

```
#topcontent {
        background-color: yellow;
}

#leftcontent {
        position: absolute;
        left:1%;
        width:20%;
        top:50px;
        background:#fff;
}

#centerleft {
        position: absolute;
        left:22%;
        width:28%;
        top:50px;
        background:#fff;
}
```

4 columns - variable width

```
#centerright {
        position: absolute;
        left:51%;
        width:28%;
        top:50px;
        background:#fff;
}

#rightcontent {
        position: absolute;
        left:80%;
        width:19%;
        top:50px;
        background:#fff;
}
```

Dreamweaver Introduction

Mehmud Abliz

Dreamweaver Basics

- Only the outline of what’ll be covered about Dreamweaver is given in slides
- Repeat the activities I demonstrated throughout the whole Dreamweaver section.

Dreamweaver Basics

- Windows and panels overview
- Menus overview
- Creating a site a Dreamweaver
- Creating a XHTML page
- Editing “Preview browser list”
- Adding text, image, tables, lists, & links
- Adding stylesheet (internal & external)

Validate your web site early

- How to validate?
  - Put the following anywhere in your code
    ```
    <a
    href="http://validator.w3.org/check?uri=[UR
    I of the web page]”> </a>
    ```
  - Use http://validator.w3.org/
  - `<a href="http://validator.w3.org/check/referer" >xhtml</a>`
  - `<a href="http://jigsaw.w3.org/css-validator/check/referer">css</a>`

Web Design & Development

Dreamweaver More Features

Mehmud Abliz

Dreamweaver

- Insert bar
- Properties inspector (contextual panel)
- Document toolbar
- Panel groups (File, tag inspector, css)
- Organizing your panels
- Site map
- Page properties (modify->page property)

- Layers
  - Is actually a absolute positioned <div> with some CSS properties

- Working with links
  - target: _blank, _parent, _self, _top
  - a:hover, a:active (A link becomes active once you click on it.), a:visited

Dreamweaver

- Linking and embedding multi-media, and setting parameters
  - Linking to an audio or video file
  - Embedding an audio or video file

- Templates
  - create a template
  - adding editable region
  - Adding repeating region
  - modify a template
- Library items
  - Create library items

Dreamweaver

- Create photo album