Web Technologies and Applications

Winter 2001

CMPUT 499: HTML and Beyond

Dr. Osmar R. Zaïane

University of Alberta

Publishing On the Web

Publishing information on the WWW is an activity that involves three major steps:

1. **Create a document**
   - HTML with any text editor
   - HTML editors
2. **Put the document on the Web**
3. **Validate the document**
   - Try the page with different browsers and Oss
   - HTML validators

Course Content

- Introduction
- Internet and WWW
- Protocols
- **HTML and beyond**
- Animation & WWW
- Java Script
- Dynamic Pages
- Perl
- Java Applets
- Databases & WWW
- SGML / XML
- Managing servers
- Search Engines
- Web Mining
- CORBA
- Security Issues
- Selected Topics
- Projects

Objectives of Lecture 4

HTML and beyond

- Get an overview of the hypertext markup language used for publishing on the WWW.
- See some advanced features of the last version of the languages such as cascading style sheets
- After the Lecture, students will demonstrate a sound understanding of HTML standards.

Outline of Lecture 4

- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

Web Publishing with HTML

- HTML stands for HyperText Markup Language.
- Language format based on SGML.
- Non-proprietary language.
- Based on a set of tags that indicate how document-content should be rendered.
- There are many versions of HTML and different browsers have their own add-ons.

History of HTML

- HTML was originally developed by Tim Berners-Lee in CERN in 1990. **HTML+** (1993) was later abandoned.
- **HTML 2.0**, now obsolete, was ratified by the IETF in 1994.
- **HTML 3.0** was a abandoned due to lack of compromise between major browsers
- **HTML 3.2** (less than HTML 3.0) is the most used standard today (since 1997)
- **HTML 4.0** (end of 1997) brought new enhancements.
- **XHTML 1.0** is the latest recommendation of the W3C. It is a combination of HTML4 and XML.

Major Syntax Components

- <**TAG** parameters=values> *text* </**TAG**>
- <**TAG** parameters=values>
- You may have many parameters or no parameters at all.
- HTML is not case sensitive
- Values can be case sensitive
- Tags for changing appearance of text, tags for structuring sections, tags for embedding objects, etc.

Do We Have to Know HTML?

- There are many HTML editors and publishing software that generate HTML.
- We can create web pages without knowing HTML.
- However, HTML editors do not generate correct HTML and the generated HTML is not easy to update.
- Many HTML editors do not deal with style sheets and JavaScript that can add neat functionalities.
- In order to write web-based application one need to know HTML since server-side scripts and programs generate HTML.

Outline of Lecture 4

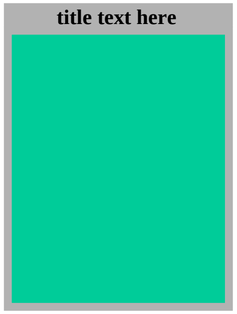
- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

Page Structure

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
   <HEAD>
        <TITLE>title text here</TITLE>
   </HEAD>

   <BODY BGCOLOR="#00FF00">

   Text and tags come here

   </BODY>
</HTML>
```

Head

Body



What Comes in the Head?

<HEAD> …… </HEAD>   Header section of the HTML page
Optional Head elements:

<TITLE> …</TITLE>	Title of page
<BASE HREF="…">	Starting location for relative URLs
<SCRIPT …>…</SCRIPT>	Embedded script programs
<STYLE …>…</STYLE>	Specify cascading style sheets
<LINK …>	Creates associations between pages
<META…>	Records document information

- **Name and Content**
  <META Name="author" Content="O. Zaiane">
- **HTTP-EQUIV**
  <META HTTP-EQUIV="Expires" Content="Sun, 31 Dec 2000 23:59">

- **And many others**

What Comes in the Body?

<BODY …> …… </BODY> **Body section of the HTML page**

Optional parameters for BODY tag:

• **BACKGROUND**	**Specifies URL of an image to display in the background**
• **BGCOLOR**	**Specifies the color of the background**
• **TEXT**	**Specifies the color of the text**
• **LINK**	**Specifies the color of the unvisited hyperlinks**
• **VLINK**	**Specifies the color of the visited hyperlinks**
• **ALINK**	**Specifies the color of the links currently selected**

```
<BODY BCKGROUND="http://www.somewhere.ca/image.gif"
    BGCOLOR="#FFFF00" TEXT="#000000" LINK="#0000FF"
    VLINK="#800000" ALINK="#008080">
…
</BODY>
```

http://developer.netscape.com/library/documentation/htmlguid/colortab.htm

Some HTML Tags

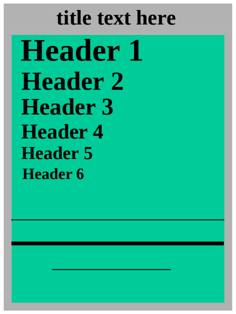
Some useful tags:

Headers:

<H1>Header 1**</H1>**
<H2>Header 2**</H2>**
<H3>Header 3**</H3>**
<H4>Header 4**</H4>**
<H5>Header 5**</H5>**
<H6>Header 6**</H6>**
<H? ALIGN=left|right|center>...</H?>

Horizontal line:

<HR>
<HR SIZE=4>
<HR WIDTH=50%>



title text here

Header 1
Header 2
Header 3
Header 4
Header 5
Header 6

Some HTML Tags – con’t

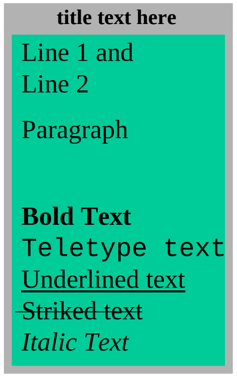
Some useful tags:

Line breaks and Paragraphs:

Line 1 and **<BR>**Line 2
<P>Paragraph**</P>**
<P ALIGN=left|right|center>...</P>
Can also use <DIV>…</DIV>
See also <NOBR>…</NOBR> and <WBR>

Bold Text and Underlined Text:

<B>Bold Text**</B>** (or <STRONG>…</STRONG>)
<TT>Teletype text**</TT>**
<U>Underlined text**</U>**
<S>Striked text**</S>** (or <STRIKE>…</STRIKE>)
<I>Italic Text**</I>**
<BLINK>blinking text**</BLINK>**



title text here

Line 1 and
Line 2

Paragraph

Bold Text
`Teletype text`
Underlined text
~~Striked text~~
Italic Text

Some HTML Tags – con’t

Some useful tags:

Descriptive markup:

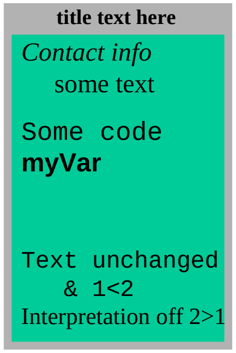
<ADDRESS>Contact info**</ADDRESS>**
<BLOCKQUOTE>some text**</BLOCKQUOTE>**
<CODE>some code**</CODE>**
<VAR>myVar**</VAR>**

Rendering depends upon browser

also <KBD>…</KBD> <DFN>…</DFN> <CITE>…</CITE> etc.

Text sections:

<PRE>Text is unchanged
 &amp; 1&lt;2**</PRE>**
<XMP>Interpretation off 2>1**</XMP>**



title text here

Contact info
some text

`Some code`
myVar

```
Text unchanged
  & 1<2
```
Interpretation off 2>1

Special characters in HTML *http://www.w3.org/pub/WWW/TR/REC-html132.html#charset*

Some HTML Tags – con’t

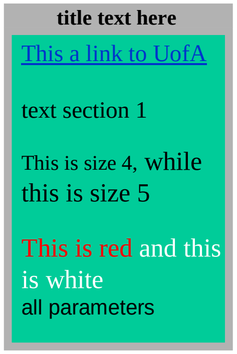
Some useful tags:

http, mailto, ftp, file, about

Hyperlinks:
<A HREF=*http://www.ualberta.ca* TARGET=_TOP>
This a link to UofA**</A>**
<A NAME=section1>text section 1**</A>**

Font sizes and Colours:
<FONT SIZE=4>this is size 4,**</FONT>**
<FONT SIZE=5>while this is size 5**</FONT>**

<FONT COLOR=red>This is red**</FONT>**
<FONT COLOR=white>and this is white**</FONT>**
<FONT COLOR=cyan SIZE=4 FACE=arial>all parameters**</FONT>**



title text here

This a link to UofA

text section 1

This is size 4, while this is size 5

This is red and this is white
all parameters

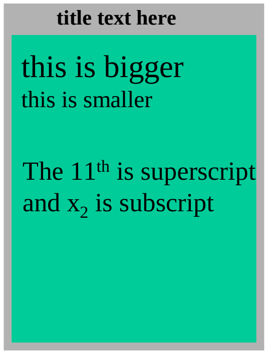
Some HTML Tags – con’t

Some useful tags:

More on sizes:
<BIG>this is bigger,**</BIG>**
<SMALL>this smaller**</SMALL>**

<BASEFONT SIZE=3>

The 11**<SUP>**th**</SUP>** is in superscript
and x**<SUB>**2**</SUB>** is subscript



title text here

this is bigger
this is smaller

The 11th is superscript and x_2 is subscript

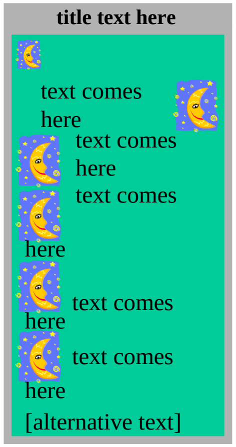
Some HTML Tags – con’t

Some useful tags:

Embedding images:
<IMG SRC=MyFile.gif **WIDTH**=100 **HEIGHT**=200>
<IMG SRC=MyFile.gif **ALIGN**=right>text comes here
<IMG SRC=MyFile.gif **ALIGN**=left>text comes here
<IMG SRC=MyFile.gif **ALIGN**=top>text comes here
<IMG SRC=MyFile.gif **ALIGN**=bottom>text comes here
<IMG SRC=MyFile.gif **ALIGN**=middle>text comes here

<IMG SRC=MyFile.gif **ALT=“*alternative text*”>**

Other parameters:
HSPACE, VSPACE, BORDER, USEMAP, ISMAP, LOWSRC, NAME



title text here

text comes here

text comes here

text comes here

text comes here

text comes here

[alternative text]

Some HTML Tags – con’t

Some useful tags:

Embedding objects:

Embedding video, audio, VRML, and other formats via Plug-ins
< EMBED SRC=“…” …> …</EMBED>
with generic attributes such as **WIDTH** and **HEIGHT** as well as plug-in-specific attributes such as **AUTOSTART**, etc.

Embedding applets
<APPLET CODE=“…” WIDTH=xxx HEIGHT=yyy …>
<PARAM NAME=“…” VALUE=“…”>
…
</APPLET>

Embedding ActiveX components
<OBJECT CLASSID=“…” …> … </OBJECT> (compromises security, good only for intranets)

Outline of Lecture 4

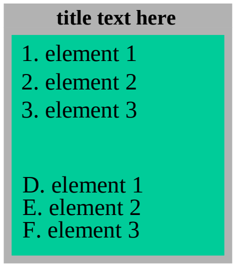
- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

Some HTML – con't

Some useful tags:

<u>Ordered lists:</u>

```
<OL>
    <LI>element 1
    <LI>element 2
    <LI>element 3
</OL>

<OL TYPE=A START=4 COMPACT>
    <LI>element 1
    <LI>element 2
    <LI>element 3
</OL>
```



title text here

1. element 1
2. element 2
3. element 3

D. element 1
E. element 2
F. element 3

TYPE:		
	1	Arabic numbers
	A	Alphabetic uppercase
	a	Alphabetic lowercase
	I	Roman numeral uppercase
	i	Roman numeral lowercase

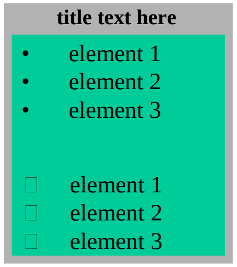
<LI TYPE=xxx VALUE=yyy>…</LI>

Some HTML – con't

Some useful tags:

<u>Unordered lists:</u>

```
<UL>
    <LI>element 1
    <LI>element 2
    <LI>element 3
</UL>

<UL TYPE=SQUARE>
    <LI>element 1
    <LI>element 2
    <LI>element 3
</UL>
```



title text here

- element 1
- element 2
- element 3

□ element 1
□ element 2
□ element 3

TYPE:		
	DISC	•
	CIRCLE	o
	SQUARE	□

<LI TYPE=xxx>…</LI>

Some HTML – con't

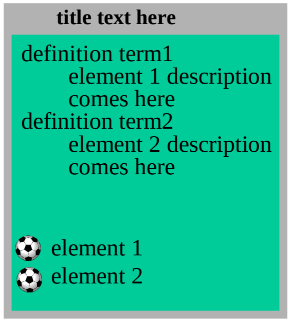
Some useful tags:

<u>Definition lists:</u>

```
<DL COMPACT>
    <DT>definition term1 </DT>
    <DD>element 1 description comes here </DD>
    <DT>definition term2 </DT>
    <DD>element 2 description comes here </DD>
</DL>
```

<u>Definition lists with image bullets:</u>

```
<DL>
    <DT><IMG SRC="soccer.gif" ALT="*"> element 1
    <DT><IMG SRC="soccer.gif" ALT="*"> element 2
</DL>
```



title text here

definition term1
        element 1 description comes here
definition term2
        element 2 description comes here

⚽ element 1
⚽ element 2

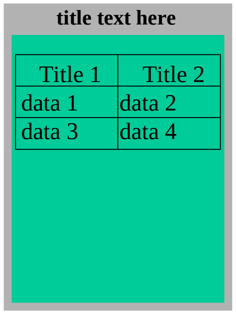
Outline of Lecture 4

- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

Basic Structure of Tables

Tables:

```
<TABLE BORDER=1 WIDTH=100%>
    <TR>
          <TH>Title1</TH><TH>Title 2</TH>
    </TR>
    <TR>
          <TD>data 1</TD><TD>data 2</TD>
    <TR>
    <TR>
          <TD>data 3</TD><TD>data 4</TD>
    </TR>
</TABLE>
```

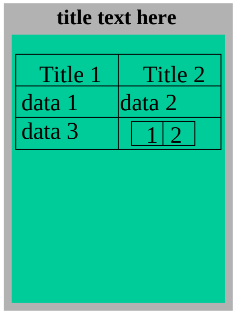


title text here

Title 1	Title 2
data 1	data 2
data 3	data 4

Tables in Tables

```
<TABLE BORDER=1 WIDTH=100%>
    <TR>
          <TH>Title1</TH><TH>Title 2</TH>
    </TR><TR>
          <TD>data 1</TD><TD>data 2</TD>
    </TR><TR>
          <TD>data 3</TD>
          <TD><TABLE><TR><TD>1</TD>
                                  <TD>2</TD>
                            </TR>
                </TABLE>
          </TD>
    </TR>
</TABLE>
```



title text here

Title 1	Title 2
data 1	data 2
data 3	1 2

Table Parameters

- BORDER
- ALIGN
- WIDTH
- CELLSPACING
- CELLPADDING
- BGCOLOR
- <CAPTION ALIGN=...>…</CAPTION>
- BORDERCOLOR (non standard)
- FRAME (non standard)

Rows and Cells Parameters

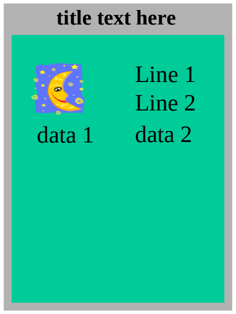
- TR
- ALIGN
- VALIGN
- BGCOLOR

- TH and TD
- COLSPAN
- ROWSPAN
- ALIGN
- VALIGN
- BGCOLOR
- WIDTH
- HEIGHT
- NOWRAP

Tables and Page Layout

Borderless tables can be useful for web page layout.

```
<TABLE BORDER=0 WIDTH=100%>
    <TR>
          <TD rowspan=2><IMG SRC=MyFile.gif>
          </TD>
          <TD> Line1</TD>
    </TR>
    <TR>
          <TD> Line 2 </TD>
    </TR>
    <TR>
          <TD>data 1</TD>
          <TD>data 2</TD>
    </TR>
</TABLE>
```



title text here

Line 1
Line 2
data 1 data 2

Outline of Lecture 4

- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

The Frames

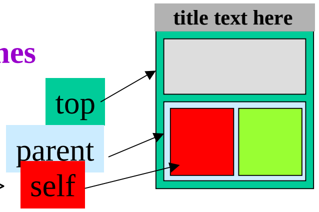
- Frames present a different template for HTML documents They were introduced as a standard in HTML 3.2.
- Rather that HEAD and BODY, the document has HEAD and FRAMESET.
- The FRAMESET divides the document in a set of frames, each referring to a URL.
- Frames contain sub documents, however, while a frame is an HTML document per se, the set of frames is also a document. (Web document view)

The Frames

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
   <HEAD>
        <TITLE>title text here</TITLE>
   </HEAD>

   <FRAMESET ROWS=50,*>
        <FRAME SRC=“MyPage.html”>
        <FRAMESET COLS=“*,*”>
               <FRAME SRC=“P1.html”>
               <FRAME SRC=“P2.html”>
        </FRAMESET>
   </FRAMESET>
   <NOFRAME>
        <BODY>You need another browser</BODY>
   </NOFRAME>
</HTML>
```

Head

Frames



The Different Framesets

<u>Having rows:</u>

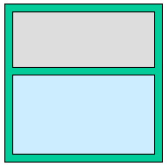
<FRAMESET ROWS=row1Size, row2Size, row3Size,…>

Size could be number of pixels, a percentage, or undetermined

<FRAMESET ROWS=50,*>

<FRAMESET ROWS=10%,*>

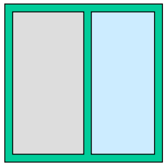
<FRAMESET ROWS=*,*> = <FRAMESET ROWS=50%,50%>



<u>Having columns:</u>

<FRAMESET COLS=col1Size, col2Size,col3Size,…>

<FRAMESET COLS=*,*>



Other parameters:

FRAMEBORDER, FRAMESPACING, BORDERCOLOR…

The Element FRAME

- A FRAME can contain another FRAMESET.
- A FRAME has a NAME and SRC.
- Other parameters: FRAMEBORDER, BORDERCOLOR, NORESIZE, SCROLLING, MARGINWIDTH, MARGINHEIGHT,…

❖ <NOFRAME>…</NOFRAME> is to display a message for browsers that do not support frames.

Targeting a Frame

- You can target a different frame to open an HTML document.

<FRAME SRC=“myPage.html **NAME=main**>

…

<A HREF=“NEWPAGE.html **TARGET=main**>

- There are predefined names *_blank*, *_top*, *_parent* and *_self*.

Why Avoid Frames?

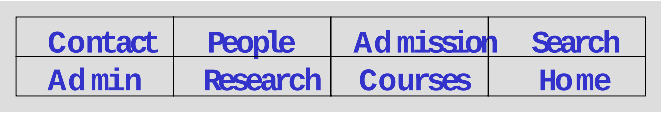
- The concept of a web document changes.
- The meaning of the “Back” and “Forward” becomes confusing to some users.
- Poorly designed frames can get the user lost and frustrated.
- It is difficult to find the URL of a HTML file contained in a frame. ➔ difficult to bookmark, print, etc.
- Some browsers do not support frames.
- Framed documents are not validated by SGML.

Outline of Lecture 4

- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

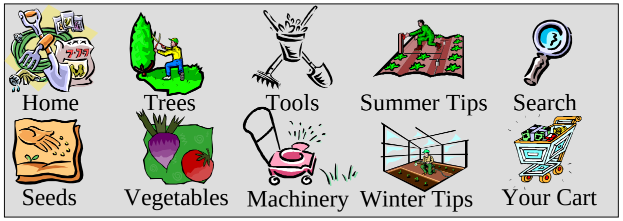
What are Image Maps?

An image-map is an image that serves as a menu: Clicking different areas lead to different URLs.



Contact	People	Admission	Search
Admin	Research	Courses	Home

Examples



Server-Side Image Map

- <A href=“mycgi.cgi”>
  <IMG SRC=“MyImage.gif” **ISMAP**>
  </A>
- ISMAP causes the image to be used as a server-side image map. The browser send the coordinates of the point clicked to the cgi on the server. The cgi determines what to do.
- Rarely used today (connection overhead).

Client-Side Image Map

- <IMG SRC=“MyImage.gif” **USEMAP=mapname**>
  <MAP NAME=mapname>…</MAP>
- The MAP tag section identifies areas that could be clicked.
- <AREA ALT=“…” HREF=“…”…>
- Parameters could be:
  - SHAPE (RECT, CIRCLE, POLY), COORDS
  - Also NOHREF

Adding Comments

- Comments can be added to an HTML file.
- Comments are ignored by the HTML interpreter.

```
<!--  These are comments -->

<!--
      These are also comments
-->
```

Outline of Lecture 4

- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

HTML Validators

- Browsers try to guess how to render pages in presence of incorrect HTML.
- Browsers don’t guess the same way.
- When there is incorrect HTML the result is unpredictable.
- With errors, some pages may still look fine with some browsers but not with others.
- Some browser specific HTML is non conform to the standards.
- W3C HTML validation service http://validator.w3.org

Outline of Lecture 4

- What is HTML?
- Basic HTML
- Building Lists
- Tables
- Frames
- Image Maps
- Validating HTML pages
- Cascading Style Sheets

What are Style Sheets?

- HTML contained tags to indicate how to render pages: tags for structure and tags for style.
- The idea of style sheets is to separate page content and page style (text vs. colour, font, format, etc.).
- Cascading Style Sheets (CSS) is a technology used as an addition to HTML and gives a sophisticated way to specify how content should be rendered (in style).

Cascade of Rules

- CSS are sets of style rules to customize HTML elements.
- Style rules can be: external (imported), inline (within an HTML element), or embedded (declared in a document).
- They could different styles applying for the same text section.
- There are general rules for determining the precedence (cascading) of the styles.

Specifying Style Rules

The syntax for specifying style properties is:

selector {property : value}

or

```
selector {property1 : value1;
          property2: value2;
          …
          propertyN : valueN}
```

Notice it is *property:value* NOT *property=value*

Style Properties

- There are many properties pertaining to:

  font, size, colour, background, margins, borders, width, height, alignment, text appearance, etc. (and even position as we shall see later)
- Not all properties are recognized by both major browsers Netscape and IE.
- Conform to W3C CSS-level1 and CSS-level2.

Style Sheets Advantages

- Separation of text content and displaying style
- Possibility to create external style templates
- Consistent rendering of style throughout site
- No need for new HTML tags for new styles
- The end of a war?

Defining Rules

- The tag <STYLE> allows the definition of formatting rules

```
<STYLE>
<!--
 Style rules
 -->
</STYLE>
```

Example

```
<STYLE>
<!--
BODY {font: 12pt Helvetica; color:blue;
margin-left: 0.5in}
H1 {font: 18pt Palatino; color: red}
H2 {font-family: MeppDisplayShadow}
KBD {text-decoration: underline}
-->
</STYLE>
```

External Style Sheets

- Style sheets can be kept separately from the HTML document.
- Possible re-use of the same style sheets with different HTML documents.
- Use the <LINK …> tag in the document HEAD.
- <LINK REL=STYLESHEET HREF="mystyle.css" TYPE="text/CSS">
- Style sheets file should be WWW-accessible.

Embedded Style Sheets

- We can add style information in the document HEAD.
- The formatting rules apply for the whole document.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML><HEAD><TITLE>Style Sheets Demo 1</TITLE>
<STYLE> <!--
  H1 {text-align:center; color:blue; font-family:Arial}
  H2, H3 {text-decoration:underline; font-style:italic}   -->
</STYLE>
</HEAD> …….
```

Inline Style Sheets

- We can create style rules within a document directly inside an HTML element tag.
- The formatting rules apply the section.
- The selector in the rule is omitted.

```
…..
<P STYLE="font: 10 pt Arial;
               line-height: 12 pt;
               margin-left: 0.5in;
               margin-right: 0.5cm;
               color: green;
               font-weight: bold">
This paragraph will be displayed as specified</P>
```

Applying the New Styles

- External and embedded style rules will automatically apply when HTML elements they are associated with are used.
- Inline style rules apply where they are defined.
- The application of the style rule on the content starts at the beginning of the opening tag and end at the closing tag.
- What about applying styles for non tag contained text?

The <SPAN> Tag

- When we want to apply a style to part of a document that is not contained between an opening and closing tag, we can use the <SPAN> … </SPAN> tag.

```
<SPAN STYLE="…">
text with new style
</SPAN>
```

```
<OL TYPE=A>
<SPAN STYLE="font-style:italic; color:red">
<LI> my first element
<LI> my second element
</SPAN>
<LI> is element is normal
</OL>
```

Example

User Defined Classes

- One can create classes of selectors associate a style to them and then use them to apply the style on sections of the document.
- For example we could define an abstract paragraph type as being in italic, 10 point text with some left and right margins: **<STYLE>**

  P.abstract {font-style:italic; font: 10pt Palatino; margin-left:0.5cm; margin-right:0.5cm} </STYLE>
- The new class is used as follows:

  <P CLASS="abstract"> this is the abstract</P>

Generic User Defined Classes

- One can also define a class that is not attached to a particular HTML element:

  .cblue {color: blue; font-weight: bold}
- The new class can be used with different HTML element or even with non contained text:

  <**H1 CLASS="cblue"**>My header is blue</**H1**>

  <**SPAN CLASS="cblue"**> this text is also blue </**SPAN**>

Cascading Rules

If multiple style rules apply to the same section of text, one rule has to be selected based on precedence:

1. Rules marked "important" have highest priority:
   - H1 {font-style: normal **!important**; color:red}
2. Author's rules have precedence over reader's rules.
3. More specific rules have precedence over less specific rules.
4. In case of a tie, the last rule specified has priority.

Positioning in HTML

- HTML allows the description of content, text images, etc, and provides some structure.
- CSS provides a wonderful way to define and maintain consistent rendering style
- Content is positioned automatically by the browser in "natural" positions.
- Frames and borderless tables help better position objects in a page.
- Difficult and limited.

CSS-P for Positioning

- LAYER and ILAYER in the early beta release of Netscape 4.0
- LAYER was rejected by W3C but still in use with Netscape
- CSSP (Cascading Style Sheets Positioning) is an extension to CSS that allows pixel-level control over the position of HTML elements.
- Based on the LAYER idea

What is a Layer?

- A Layer is an area that may contain text or objects and different layers can overlap.
- Layers are implemented differently in IE and Netscape.
- What follows is cross-browser valid
- Do not use LAYER tag, use DIV tag
- DIV tag, DIV element, DIV block, CSS-layer and LAYER are all synonyms and simply refer to the same thing: a layer.

```
<DIV>
   this is a
  division
</DIV>
```

```
<DIV ID=“mydiv”>
   this is a
   named division
</DIV>
```

Layer with inline CSS

```
<DIV ID=“mydiv” STYLE=“styles here”>
   this is a division with style
</DIV>
```

Cross-Browser CSS Properties

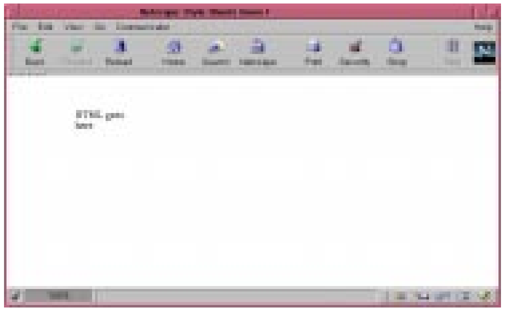
- Position [relative | absolute]
- Left [location in pixels]
- Top [location in pixels]
- Width
- Height
- Clip [defines a clipping rectangle]
- Visibility [visible | hidden | inherit]
- Z-index [staking order of layers]
- Background-color (and layer-background-color)
- Background-image (and layer-background-image)

Layer with SYTLE tag

```
<STYLE TYPE=“text/CSS”>
<! --
  #mydiv {styles here}
--> </STYLE>
<DIV ID=“mydiv”>
    this is a division with style
</DIV>
```

Example 1: Content Positioning

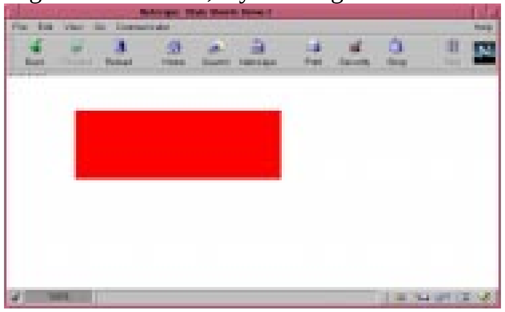
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Style Sheets Demo 1</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF">
<DIV ID="mydiv" STYLE="position:absolute; left:100; top:50; width:80;">
HTML goes here
</DIV>
</BODY>
</HTML>
```



HTML goes here

Example 2: Content Positioning

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD><TITLE>Style Sheets Demo 2</TITLE></HEAD>
<BODY BGCOLOR="#FFFFFF">
<DIV ID="mydiv" STYLE="position:absolute; left:100; top:50; width:300;
height:100; clip:rect(0,300,100,0); background-color:red; layer-background-
color:red;">
</DIV>
</BODY>
</HTML>
```



What's Next?

- After seeing HTML, the Cascading Style Sheets and the positioning capabilities, the next step is dynamic web pages, or **Dynamic HTML**.
- Dynamic HTML is a term used to describe HTML pages with dynamic content.
- There are three components in dynamic HTML:
  1. HTML
  2. Cascading Style Sheets (CSS)
  3. JavaScript
- The three components are glued together with **DOM**, the Document Object Model.