core WEB programming

Cascading Style Sheets

Agenda

- Specifying style sheet rules
- External and inline style specifications
- Creating new HTML elements through style sheet classes
- Specifying font and text properties
- Controlling foreground and background properties
- Netscape LAYERs
- Creating layers through style sheets

Benefits of Cascading Style Sheets

- **Powerful and flexible way to specify the formatting of HTML elements**
  - Can define font, size, background color, background image, margins, etc.
- **Share style sheets across multiple documents or entire Web site**
- **Can specify a class definition for a style, effectively defining new HTML elements**
- **Rules are applied in a hierarchical manner (precedence rules)**

Cascading Style Sheets

- **CSS, Level 1 (1996)**
  - Concerned with applying simple styles to HTML elements
  - http://www.w3.org/TR/REC-CSS1
- **CSS, Level 2 (1998)**
  - Supports media-specific style sheets (visual browsers, aural devices, printers, braille devices)
  - http://www.w3.org/TR/REC-CSS2
- **CSS, Level 3 (draft 2001)**
  - Focused on modularization of the CSS specification
  - http://www.w3.org/TR/css3-roadmap/
- **Note:**
  - CSS1 is supported by Netscape and Internet Explorer 4.x and above
  - See http://www.webreview.com/style/css1/charts/mastergrid.shtml for a summary of browser compatibility

Specifying Style Rules

- **General form of rule**

```
selector { property: value }
```

or

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

- **Example**

```
H1 { text-align: center;
     color: blue }
```

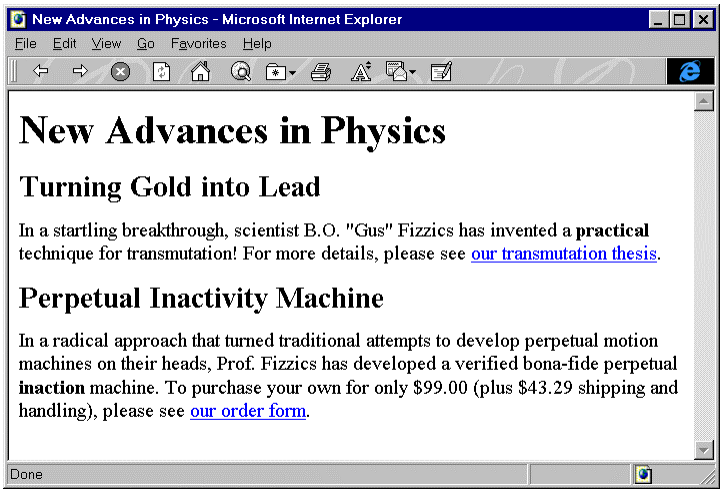

Fizzics1.html, Example (no style sheet)

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>New Advances in Physics</TITLE>
</HEAD>
<BODY>
<H1>New Advances in Physics</H1>

<H2>Turning Gold into Lead</H2>
In a startling breakthrough, scientist B.O. "Gus" Fizzics
has invented a <STRONG>practical</STRONG> technique for
transmutation! For more details, please see
<A HREF="give-us-your-gold.html">our transmutation thesis</A>.
...

</BODY>
</HTML>
```

Fizzics1.html, Result (no style sheet)

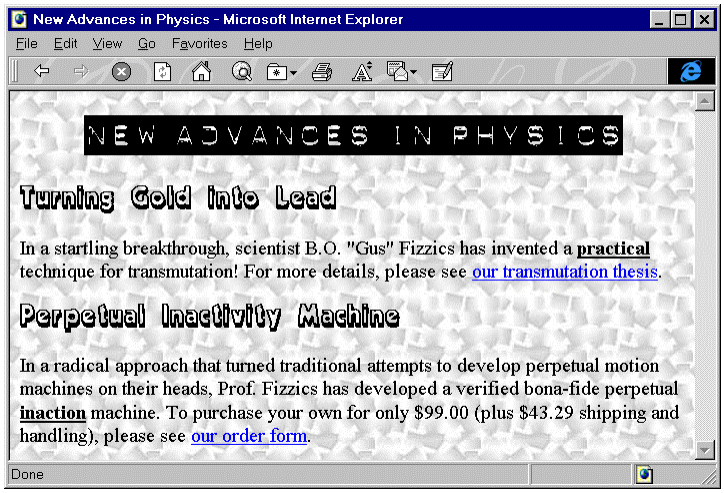


Fizzics2.html, Example (with style sheet)

- **Style information**

```
<HEAD>
  <TITLE>Document Title</TITLE>
  <STYLE TYPE="text/css">
  <!--
  BODY { background: URL(images/confetti-background.jpg) }
  H1 { text-align: center;
       font-family: Blackout }
  H2 { font-family: MeppDisplayShadow }
  STRONG { text-decoration: underline }
  -->
  </STYLE>
</HEAD>
```

Fizzics2.html, Result (with style sheet)



External Style Sheets

- Specify link to external style sheet in the HEAD section of the HTML document

```
<LINK REL=STYLESHEET
      HREF="Sitestyle.css" // Absolute or relative link
      TYPE="text/css">
```

- Sitestyle.css

```
/* Example of an external style sheet */

H1 { text-align: center;
     font-family: Arial
}
H2 { color: #440000;
     text-align: center;
     font-family: Arial Black, Arial, Helvetica, sans-serif
}

...
```

Inline Style Specification

- Use the STYLE attribute defined for each HTML element to directly specify the style
- Example

```
...
<H1>New Advances in Physics</H1>
<P STYLE="margin-left: 0.5in;
          margin-right: 0.5in;
          font-style: italic">
This paper gives the solution to three
previously unsolved problems: turning lead into gold,
antigravity, and a practical perpetual motion machine.
...
```


Defining Style Classes

- To define an *element* style class proceed the HTML element by a period and class name

```
// Define an "abstract" paragraph type
P.abstract { margin-left: 0.5in;
             margin-right: 0.5in;
             font-style: italic }
```

- To use, supply the name of the style class in the CLASS attribute of the HTML element

```
<H1>New Advances in Physics</H1>
<P CLASS="abstract">
This paper gives the solution to three previously
unsolved problems: turning lead into gold,
antigravity, and a practical perpetual motion machine.
```

Defining Style Classes

- To define a *global* style class, omit the element name

```
// Style available to all elements
.blue { color: blue; font-weight: bold }
```

- To use, simple specify the style class in the CLASS attribute of the HTML element

```
<H2 CLASS="blue">A Blue Heading</H2>

<!-- Apply to a section of text -->
This text is in the default color, but
<SPAN CLASS="blue">this text is blue.</SPAN>
```


Defining Styles through User-Defined IDs

- An ID is like a class but can be applied only once in a document

```
<HEAD>
<TITLE>...</TITLE>
<STYLE TYPE="text/css">
<!--
#foo { color: red }
-->
</STYLE>
</HEAD>
<BODY>
...
<P ID="foo">
...
</BODY>
```

Style Sheet Precedence Rules

1. **Rules marked "important" have the highest priority (rarely used)**

```
H1 { color: black !important;
     font-family: sans-serif }
```

2. **Author rules have precedence over reader rules**
   - Style sheet rules override browser preferences

Style Sheet Precedence Rules, cont.

3. **More specific rules have precedence over less specific rules**

```
#foo { ... }       // ID selector highest priority
P.big H1 { ... }   // Class higher over element
P STRONG { ... }   // Two tags higher than single tag
STRONG { ... }
```

4. **In case of tie, the last rule has priority**

Useful Font Properties

- **font-weight**
  - Relative weight (boldness) of font
  - **normal** | lighter | bold | bolder | 100 | 200 | ... | 900

```
H1 { font-weight : 200 }
H2 { font-weight : bolder }
```

- **font-style**
  - Font face type within a family
  - **normal** | italic | oblique

```
P  { font-style : normal }
TH { font-sytle : italic }
```


Useful Font Properties, cont.

- **font-size**
  - Either relative or absolute size of font
  - pt, pc, in, cm, mm | em, ex, px, % | xx-large | x-large | large | **medium** | small | x-small | xx-small | smaller | larger

```
STRONG { font-size: 150% }
P { font-size: 14pt }
P { font-size: xx-large }
```

- **font-family**
  - Typeface family for the font

```
H1 { font-family: Arial }
```

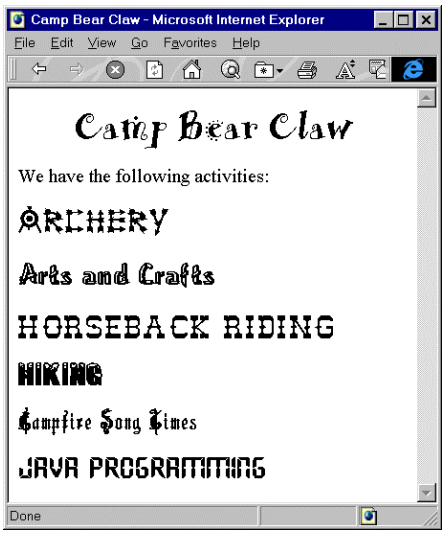
CampBearClaw.html, Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Camp Bear Claw</TITLE>
  <LINK REL=STYLESHEET HREF="CampBearClaw.css" TYPE="text/css">
</HEAD>
<BODY>
<H1>Camp Bear Claw</H1>
We have the following activities:
<H2 CLASS="archery">Archery</H2>
<H2 CLASS="arts">Arts and Crafts</H2>
<H2 CLASS="horseback">Horseback Riding</H2>
<H2 CLASS="hiking">Hiking</H2>
<H2 CLASS="campfire">Campfire Song Times</H2>
<H2 CLASS="java">Java Programming</H2>
</BODY>
</HTML>
```


CampBearClaw.css

```
H1 { text-align: center;
     font-family: Funstuff }
H2.archery { font-family: ArcheryDisplay }
H2.arts { font-family: ClampettsDisplay }
H2.horseback { font-family: Rodeo }
H2.hiking { font-family: SnowtopCaps }
H2.campfire { font-family: Music Hall }
H2.java { font-family: Digiface }
```

CampBearClaw.html, Result



Useful Text Properties

- **text-decoration**
  - Describes text additions or "decorations" that are added to the text of an element
  - **none** | underline | overline | line-through | blink

```
P { text-decoration: underline }
```

- **vertical-align**
  - Determines how elements are positioned vertically
  - top | bottom | **baseline** | middle | sub | super | text-top | text-bottom | %
- **text-align**
  - Determines how paragraphs are positioned horizontally
  - **left** | right | center | justify

Useful Text Properties, cont.

- **text-indent**
  - Specifies the indentation of the *first* line of the paragraph
  - +/– pt, pc, in, cm, mm | +/– em, ex, px, %

```
P { text-indent: -25px } /* Hanging indent */
```

- **line-height**
  - Specifies the distance between two consecutive baselines in a paragraph
  - **normal** | number | pt, pc, in, cm, mm | em, ex, px, %

```
.double { line-height: 200% }
.triple { line-height: 3 } /* 3x the font size */
DIV { line-height: 1.5em }
```

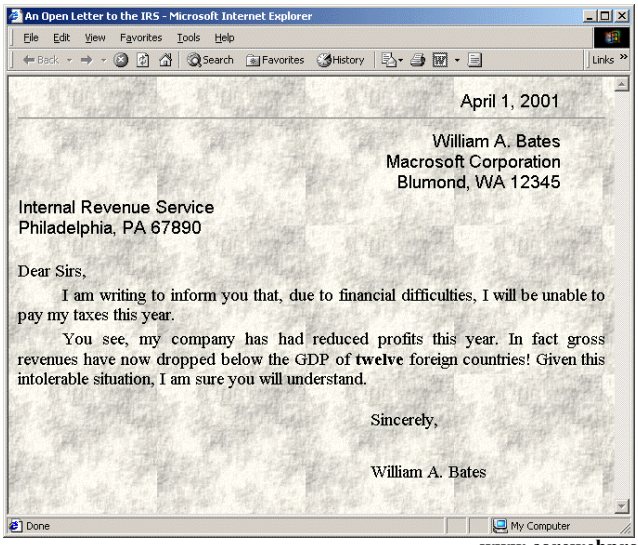

Bates.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>An Open Letter to the IRS</TITLE>
  <LINK REL=STYLESHEET HREF="Bates.css" TYPE="text/css">
</HEAD>
<BODY BACKGROUND="images/bond-paper.jpg">
<P CLASS="rhead">
April 1, 2001
<HR>
<P CLASS="rhead">
William A. Bates<BR>
Macrosoft Corporation<BR>
Blumond, WA 12345
<P CLASS="lhead">
Internal Revenue Service<BR>
Philadelphia, PA 67890
<P>
<BR>
Dear Sirs,
<P CLASS="body">
I am writing to inform you that, due to financial difficulties,
...
```

Bates.css

```
P { margin-top: 5px }
P.rhead { text-align: right;
          margin-right: 0.5in;
          font-family: sans-serif }
P.lhead { font-family: sans-serif }
P.body { text-align: justify;
         text-indent: 0.5in }
P.foot { margin-left: 60%;
         line-height: 300% }
```


Bates.html



Useful Foreground and Background Properties

- **color**
  - Color of the text or foreground color
  - color-name | #RRGGBB | #RGB | rgb(rrr, ggg, bbb) | rgb(rrr%, ggg%, bbb%)

```
P  { color : blue }
H1 { color : #00AABB }
H3 { color : rgb(255, 0, 0 ) } /* red */
```

- **background-image**
  - **none** | url(*filename*)
  - Specifies an image to use as the background of region

```
H2 { background-image: url(Bluedrop.gif);}
```


Useful Foreground and Background Properties, cont.

- **background-repeat**
  - Specifies how to tile the image in the region
  - **repeat** | repeat-x | repeat-y | norepeat

```
BODY {
  background-image: url(Bluedot.gif);
  background-repeat: repeat-x;
}
```

- **background**
  - Lets you combine properties in a single entry

```
P { background: url(wallpaper.jpg) repeat-x }
```

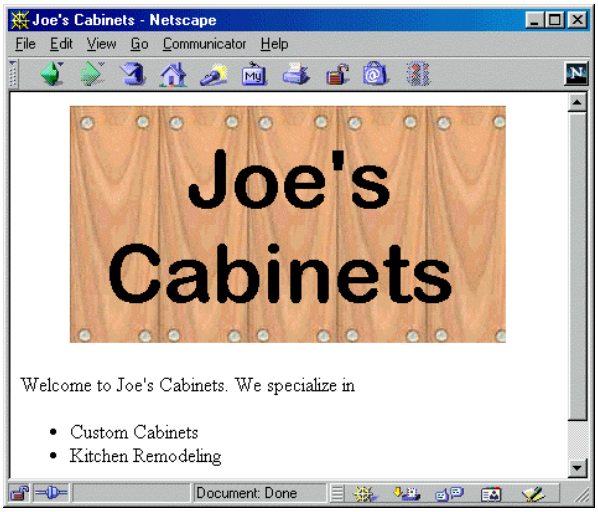
Cabinets.html, Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Joe's Cabinets</TITLE>
  <LINK REL=STYLESHEET HREF="Cabinets.css" TYPE="text/css">
</HEAD>
<BODY>
<CENTER>
<TABLE WIDTH=360 HEIGHT=199>
  <TR><TD ALIGN="CENTER" CLASS="banner">Joe's Cabinets
</TABLE>
</CENTER>
<P>
Welcome to Joe's Cabinets. We specialize in
<UL>
  <LI>Custom Cabinets
  <LI>Kitchen Remodeling
  <!-- Etc -->
</UL>
<!-- Etc -->
</BODY>
</HTML>
```


Cabinets.css

```
.banner { background: url(images/boards.jpg) repeat-x;
          font-size: 50pt;
          font-family: Arial Rounded MT Bold }
```

Cabinets.html, Result

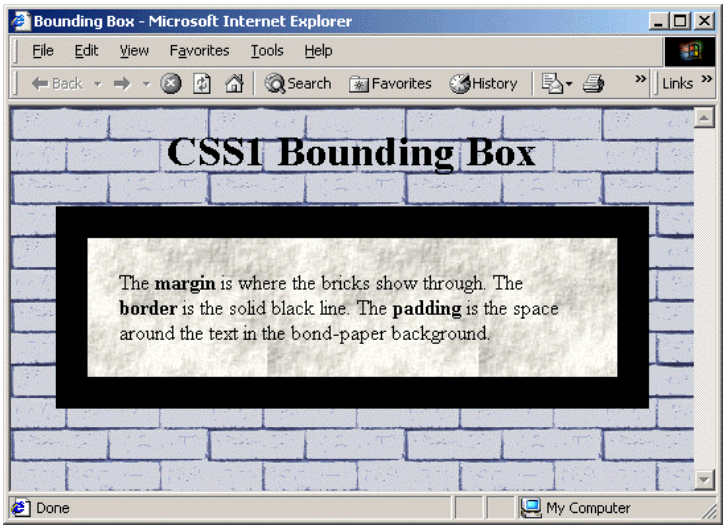


Properties of the Bounding Box

- CSS assume that all elements result in one or more rectangular regions (bounding box)
- Styles can specify the margins, borders, and padding of the bounding box

```
P { margin: 0.25in;
    border: 0.25in solid black;
    padding: 0.25in;
    background: URL(images/bond-paper.jpg) }
```

The Bounding Box



Images and Floating Elements

- **width, height**
  - Specify a fixed size for an element (usually an image)
  - **auto** | pt, pc, in, cm, mm | em, ex, px

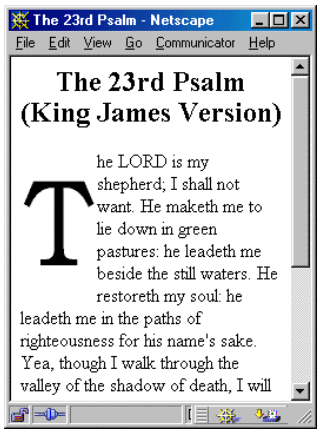
```
IMG.bullet { width: 50px; height: 50px }
```

- **float**
  - This property lets elements float into the left or right margins where the text wrapping arounds
  - **none** | left | right

Psalm23.html

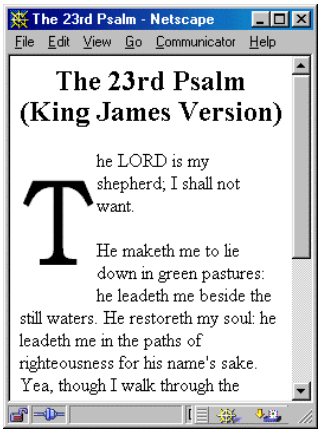
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>The 23rd Psalm</TITLE>
<STYLE>
<!--
SPAN { float: left;
       font-family: "Cushing Book";
       font-size: 75pt }
-->
</STYLE>
</HEAD>
<BODY>
<H2 ALIGN="CENTER">
The 23rd Psalm (King James Version)</H2>
<SPAN>T</SPAN>he LORD is my shepherd; I shall not want.
He maketh me to lie down in green pastures: he leadeth me
beside the still waters. He restoreth my soul: he leadeth me
in the paths of righteousness for his name's sake. Yea,
```


Psalm23.html, Result

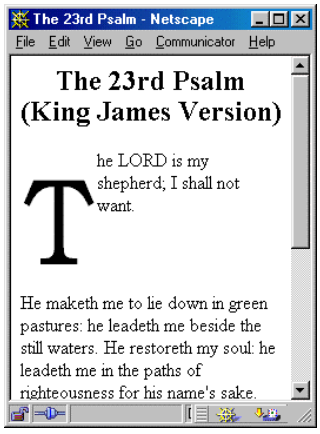


The float property can be used to implement “drop caps”

Effect of Paragraph Breaks, Example, cont.



Adding a `<P>` element simply continues the flow



Adding `<P STYLE="clear: left">` forces the next paragraph to start after the floating element

Netscape Layers

- **Layers (LAYER and ILAYER elements) allow you to:**
  - Place HTML markup in separate regions
  - Position each region on the page
- **LAYER and ILAYER are only supported in Netscape 4**
  - Layers are not supported in Internet Explorer or Netscape 6

LAYER and ILAYER

- **LAYER element**
  - The LAYER element creates regions that have an absolute position with respect to the window or parent layer
- **ILAYER**
  - The ILAYER element creates inline layers (regions that are embedded in the flow of the text)

LAYER, Example

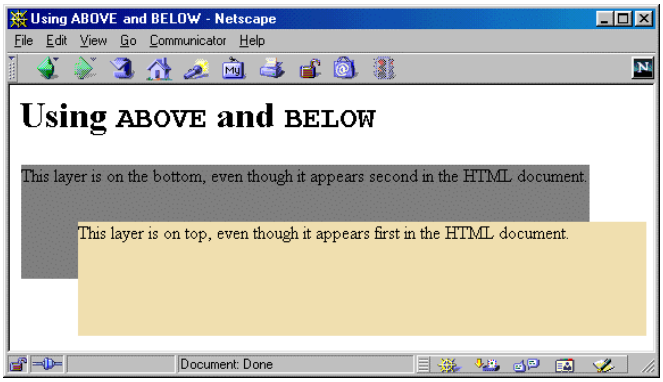
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Using ABOVE and BELOW</TITLE>
</HEAD>
<BODY>
<H1>Using <CODE>ABOVE</CODE> and <CODE>BELOW</CODE></H1>

<LAYER ID="Top" LEFT=60 TOP=120
       WIDTH=500 HEIGHT=100 BGCOLOR="#F5DEB3">
This layer is on top, even though it appears
first in the HTML document.
</LAYER>

<LAYER ID="Bottom" ABOVE="Top" LEFT=10 TOP=70
       WIDTH=500 HEIGHT=100 BGCOLOR="gray">
This layer is on the bottom, even though it appears
second in the HTML document.
</LAYER>

</BODY>
</HTML>
```

LAYER, Result

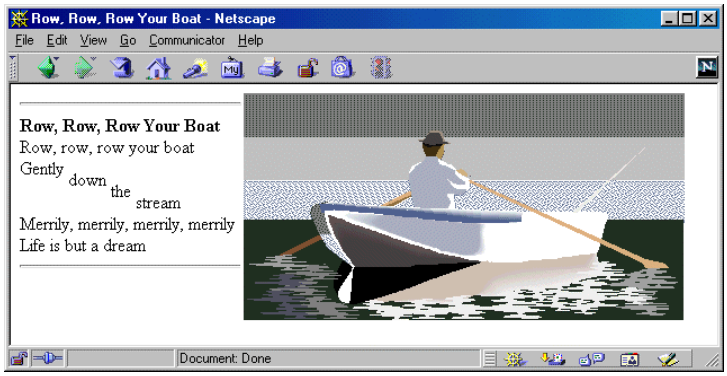


Netscape 4 layers can specify the relative order of the layers.

ILAYER, Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Row, Row, Row Your Boat</TITLE>
</HEAD>
<BODY>
<IMG SRC="images/Rowboat.gif" ALIGN="RIGHT">
<HR>
<B>Row, Row, Row Your Boat</B><BR>
Row, row, row your boat<BR>
Gently
<ILAYER TOP=10>down</ILAYER>
<ILAYER TOP=20>the</ILAYER>
<ILAYER TOP=30>stream<BR>
Merrily, merrily, merrily, merrily<BR>
Life is but a dream<BR>
<HR>
</ILAYER>
</BODY>
</HTML>
```

ILAYER, Result



Using `TOP` in `ILAYER` can move text up or down in the current paragraph.

Specifying Layers with Style Sheets

- **Style sheets provide an alternative to LAYER and ILAYER elements**
  - Style sheet layers are supported by *both* Netscape and Internet Explorer
  - However, Netscape layers are more complete
    - No equivalent style for `PAGEX` and `PAGEY` for positioning
- **Problem**
  - Netscape and IE use a different object model to refer to layers
  - See http://www.stopbadtherapy.com/standards.shtml for creating cross-browser layers

Creating a Style Sheet Layer

- **Use an ID tag format to define a style**

```
#layer1 { position: absolute;
          left: 50px; top: 75px;
          ... }
```

- **Define the layer through a DIV or SPAN element**

```
<SPAN ID="layer1">
...
</SPAN>

<DIV ID="layer2">
...
</DIV>
```


Useful Layer Properties

- **left, top**
  - Specifies the left and top sides of the layer relative to the parent window
- **position**
  - Describes how the position is defined to the parent window
  - absolute, relative, static
- **visibility**
  - Determines whether a layer is visible or hidden
  - visible, hidden, inherit

Dynamically Changing a Layer's Visibility, Example

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
  <TITLE>Changing Visibility Dynamically</TITLE>
<STYLE>
<!--
#layer1 { position: absolute; left: 0.25in; top: 1.5in;
          color: black; background-color: #F5DEB3;
          visibility: hidden }
#layer2 { position: absolute; left: 0.25in; top: 1.5in;
          color: #F5DEB3; background-color: black;
          visibility: hidden }
H1 { text-align: center;
     font-family: Arial }
FORM { text-align: center }

-->
</STYLE>
...
```


Dynamically Changing a Layer's Visibility, Example

```
<SCRIPT TYPE="text/javascript">
<!--
function display(value1, value2){
  if(document.layers) { // Test for Netscape.
    document.layers.layer1.visibility = value1;
    document.layers.layer2.visibility = value2;
  } else {
    document.all.layer1.style.visibility = value1;
    document.all.layer2.style.visibility = value2;
  }
}
//-->
</SCRIPT>
</HEAD>
<BODY BGCOLOR="WHITE">
<H1>Changing Visibility Dynamically</H1>
<FORM>
  <INPUT TYPE="BUTTON" VALUE="Show Layer1"
         onClick="display('visible', 'hidden')">
  <INPUT TYPE="BUTTON" VALUE="Show Layer2"
         onClick="display('hidden', 'visible')">
</FORM>
...
```

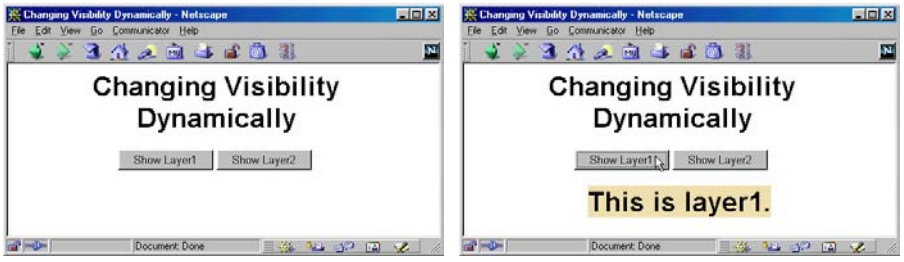
Dynamically Changing a Layer's Visibility, Example

```
...
<DIV ID="layer1">
<H1>This is layer1.</H1>
</DIV>

<DIV ID="layer2">
<H1>This is layer2.</H1>
</DIV>

</BODY>
</HTML>
```


Dynamically Changing a Layer's Visibility, Result



Selecting a button displays a hidden layer.

Appendix, Length Units

Unit	Description
cm	Centimeters (absolute unit)
em	The height of the current font (relative unit)
ex	The height of the letter “x” in the current font (relative unit)
in	Inches (absolute unit)
mm	Millimeters (absolute unit)
pc	Picas; 6 picas per inch; 12 points per pica (absolute unit)
pt	Points; 72 points per inch (absolute unit)
px	Pixels (relative unit)

Summary

- Through style sheets you can specify the general formatting of HTML elements
- Use external style sheets to share styles across all documents in the Web site
- Class definitions allow you to define multiple styles for an HTML element
- LAYERs are only supported by Netscape 5; however, a viable alternative are style sheet layers

core WEB programming

Questions?