

## *Cascading Style Sheets*

## *Outline*

- 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

CS 4390 Web Programming

Cascading Style Sheets

2

## *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)

CS 4390 Web Programming

Cascading Style Sheets

3

## *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>

CS 4390 Web Programming

Cascading Style Sheets

4

## *Cascading Style Sheets*

- 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

CS 4390 Web Programming

Cascading Style Sheets

5

## *Specifying Style Rules*

- General form of rule
  - Selector
  - Declaration -- property : value

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

OR

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

Example

```
H3 { text-align: center;  
    color: blue }
```

CS 4390 Web Programming

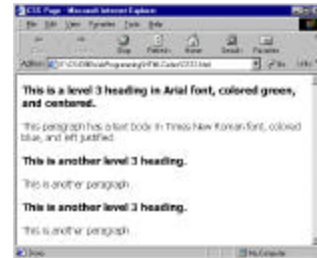
Cascading Style Sheets

6

## Example with no style sheet

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<HEAD>
<TITLE> CSS Page </TITLE>
</HEAD>
<BODY>
<H3>This is a level 3 heading in Arial font, colored green, and
centered.</H3>
<P>This paragraph has a text body in Times New Roman font,
colored blue, and left justified.
<H3>This is another level 3 heading.</H3>
<P>This is another paragraph.
<H3>This is another level 3 heading.</H3>
<P>This is another paragraph.
</BODY>
</HTML>
```

## Example with no style sheet

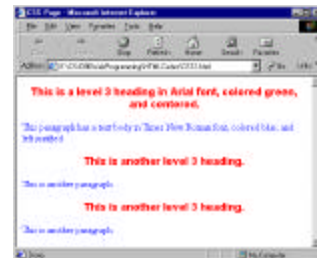


## Example with style sheet

- Style information

```
<HEAD>
<TITLE>CSS Page</TITLE>
<STYLE TYPE="text/css">
<!--
H3 { font-family: arial, san-serif;
color: red;
text-align: center }
P { font-family: "times new roman, serif;
color: blue;
text-align: center; }
-->
</STYLE>
</HEAD>
```

## Example with style sheet



## External Style Sheets

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

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

- Style.css

```
/* Example of an external style sheet */
H1, H2, H3 { font-family: arial, san-serif;
color: red;
text-align: center }
P { font-family: "times new roman", serif;
color: blue;
text-align: left } ...
```

## 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.

```
...
```

## Selectors

There are 4 most common selector types

- HTML elements
- User-defined classes
- User-defined IDs
- Anchor pseudoclasses

## HTML Elements

- Any HTML element can be used a selector
- Text elements inherit style rules from **body** selector
- Contextual selector -- nested or multiple selectors

```
H1 EM {color : green}
P b {color : blue}
```

## 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 }
```

## Defining Style Classes

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

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

<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

CS 4390 Web Programming

Cascading Style Sheets

19

## Style Sheet Precedence Rules,

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

CS 4390 Web Programming

Cascading Style Sheets

20

## Useful Font Properties

- font-weight
    - Relative weight (boldness) of font
    - **normal** | lighter | bold | bolder | 100 | 200 | ... | 900
  - font-style
    - Font face type within a family
    - **normal** | italic | oblique
- ```
P { font-style : normal }
TH { font-sytle : italic }
```

CS 4390 Web Programming

Cascading Style Sheets

21

## Useful Font Properties

- 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
- font-family
  - Typeface family for the font

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

CS 4390 Web Programming

Cascading Style Sheets

22

## 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>
```

CS 4390 Web Programming

Cascading Style Sheets

23

## CampBearClaw.css

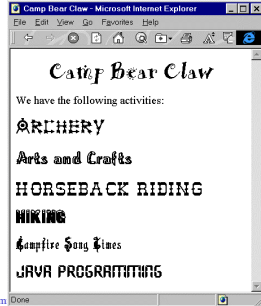
```
H1 { text-align: center;
    font-family: Funstuff }
H2.archery { font-family: ArcheryDisplay }
H2.arts { font-family: ClampettsDisplay }
H2.horseback { font-family: Rodeo }
H2.hiking { font-family: SnowtopCaps }
H2.campfire { font-family: Music Hall }
H2.java { font-family: Digiface }
```

CS 4390 Web Programming

Cascading Style Sheets

24

## CampBearClaw.html, Result



CS 4390 Web Program

25

## 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 | %

CS 4390 Web Programming

Cascading Style Sheets

26

## Useful Text Properties

- text-align
  - Determines how paragraphs are positioned **horizontally**
  - **left** | right | center | justify
- 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 */
```

CS 4390 Web Programming

Cascading Style Sheets

27

## 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)

CS 4390 Web Programming

Cascading Style Sheets

28

## Useful Text Properties

- 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 }
```

CS 4390 Web Programming

Cascading Style Sheets

29

## 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="thead">
April 1, 2001
</P>
<P CLASS="thead">
William A. Bates<BR>
Macrossoft Corporation<BR>
Blumond, WA 12345
</P CLASS="thead">
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,
...
```

CS 4390 Web Programming

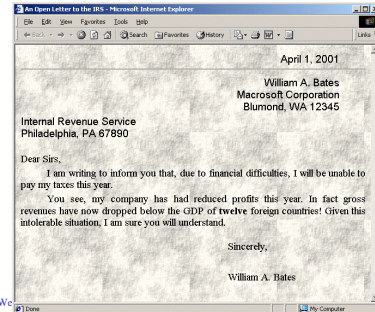
Cascading Style Sheets

30

## 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

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

```
BODY {
    background-image: url(Bluedrop.gif);
    background-repeat: repeat-x;
}
```
- background
  - Lets you combine properties in a single entry

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

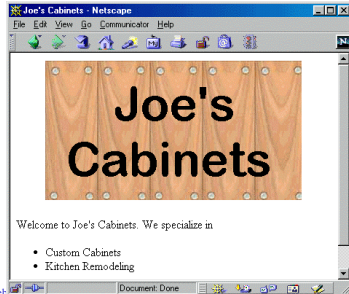
## Cabinets.html, Example

```
<!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
    </TD>
    </TR>
    </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



CS 4390 Web

37

## 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

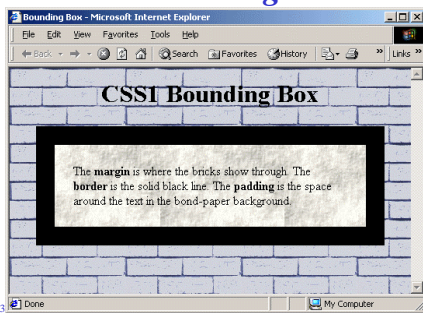
```
P { margin: 0.25in;
    border: 0.25in solid black;
    padding: 0.25in;
    background: URL(images/bond-paper.jpg) }
```

CS 4390 Web Programming

Cascading Style Sheets

38

## The Bounding Box



CS 43

39

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

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

CS 4390 Web Programming

Cascading Style Sheets

40

## Psalm23.html

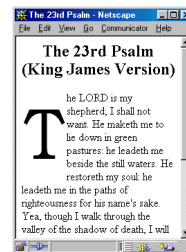
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE>The 23rd Psalm</TITLE>
<STYLE>
<!--
SPAN { float: left;
      font-family: "Dashing Book";
      font-size: 75pt }
-->
</STYLE>
</HEAD>
<BODY>
<H2 ALIGN="CENTER">
The 23rd Psalm (King James Version)</H2>
<SPAN>T</SPAN>he 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,
```

CS 4390 Web Programming

Cascading Style Sheets

41

## Psalm23.html, Result



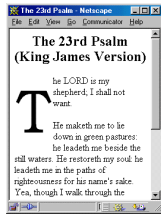
The float property can be used to implement "drop caps"

CS 4390 Web Programming

Cascading Style Sheets

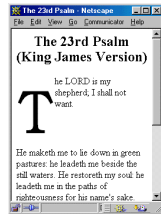
42

## Effect of Paragraph Breaks, Example, cont.



Adding a <P> element simply continues the flow

CS 4390 Web Programming



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

Cascading Style Sheets

43

## Netscape Layers

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

CS 4390 Web Programming

Cascading Style Sheets

44

## LAYER and IFRAME

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

CS 4390 Web Programming

Cascading Style Sheets

45

## LAYER, Example

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

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

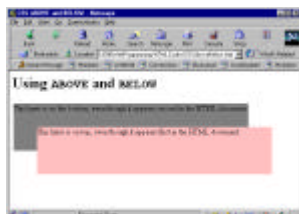
<LAYER ID="Bottom" ABOVE="Top" LEFT=10 TOP=70
  WIDTH=500 HEIGHT=100 BACKGROUND="gray">
  This layer is on the bottom, even though it appears
  second in the HTML document.
</LAYER>
</BODY>
</HTML>
```

CS 4390 Web Programming

Cascading Style Sheets

46

## LAYER, Result



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

CS 4390 Web Programming

Cascading Style Sheets

47

## IFRAME, 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">
<BR>
<B>Row, Row, Row Your Boat</B><BR>
  Row, row, row your boat<BR>
  Gently
  <LAYER TOP=10>down</IFRAME>
  <LAYER TOP=10>the</IFRAME>
  <LAYER TOP=10>stream<BR>
  Merrily, merrily, merrily, merrily<BR>
  Life is but a dream<BR>
</B>
</IFRAME>
</BODY>
</HTML>
```

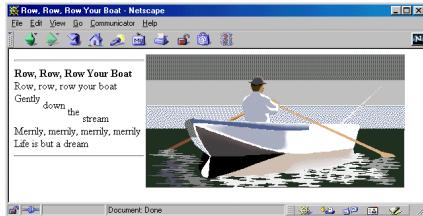
CS 4390 Web Programming

Cascading Style Sheets

48



## ILAYER, Result



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

CS 4390 Web Programming

Cascading Style Sheets

49

## 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 PEX and PEXY 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

CS 4390 Web Programming

Cascading Style Sheets

50

## 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>
```

CS 4390 Web Programming

Cascading Style Sheets

51

## 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

CS 4390 Web Programming

Cascading Style Sheets

52

## 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>
...
</HTML>
```

CS 4390 Web Programming

Cascading Style Sheets

53

## 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>
<!--
<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>
...
</BODY>
```

CS 4390 Web Programming

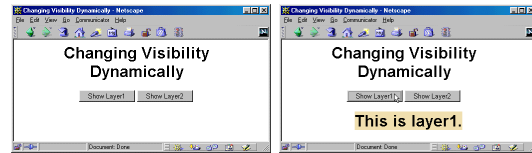
Cascading Style Sheets

54

## *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.

## *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