Tutorial 3

Working with Cascading Style Sheets

Objectives

- Review the history and concepts of CSS
- Explore inline styles, embedded styles, and external style sheets
- Understand style precedence and style inheritance
- Understand the CSS use of color
- Explore CSS styles for fonts and text
- Review and compare different image formats

Objectives

- Display an animated graphic
- Apply a background image to an element
- Float elements on a Web page
- Explore the properties of the box model
- Apply border styles to an element

Introducing Cascading Style Sheets

- **Style sheets** are declarations that describe the layout and appearance of a document
- **Cascading Style Sheets** (**CSS**) is a style sheet language used on the Web
  - CSS specifications are maintained by the World Wide Web Consortium (W3C)
  - Several versions of CSS exist: CSS1, CSS2, CSS 2.1, and CSS3

Cascading Style Sheets

- **CSS1** introduced styles for the following document features:
  - Fonts
  - Text
  - Color
  - Backgrounds
  - Block-level Elements

Cascading Style Sheets

- **CSS2** introduced styles for the following document features:
  - Positioning
  - Visual Formatting
  - Media Types
  - Interfaces
- **CSS 2.1** did not add any new features to the language

Cascading Style Sheets

- **CSS3** (which is still in development) will introduce styles for the following document features:
  - User Interfaces
  - Accessibility
  - Columnar layout
  - International Features
  - Mobile Devices
  - Scalable Vector Graphics

Applying a Style Sheet

- Three ways to apply a style to an HTML or XHTML document:
  - **Inline Styles**
  - **Embedded Style Sheet**
  - **External Style Sheet**

Using Inline Styles

- Inline styles are easy to use and interpret because they are applied directly to the elements they affect

```
<element style="style1: value1; style2:
value2; style3: value3;…">
```

Using Embedded Styles

- You can embed style definitions in a document head using the following form:

```
<style type="text/css">
   style declarations
</style>
```

- Where *style declarations* are the declarations of the different styles to be applied to the document

Using an External Style Sheet

- Because an embedded style sheet only applies to the content of one file, you need to place a style declaration in an **external style sheet** to apply to the rest of the Web site
- An **external style sheet** is a text file that contains style declarations
  - It can be linked to any page in the site, allowing the same style declaration to be applied to the entire site

Using an External Style Sheet

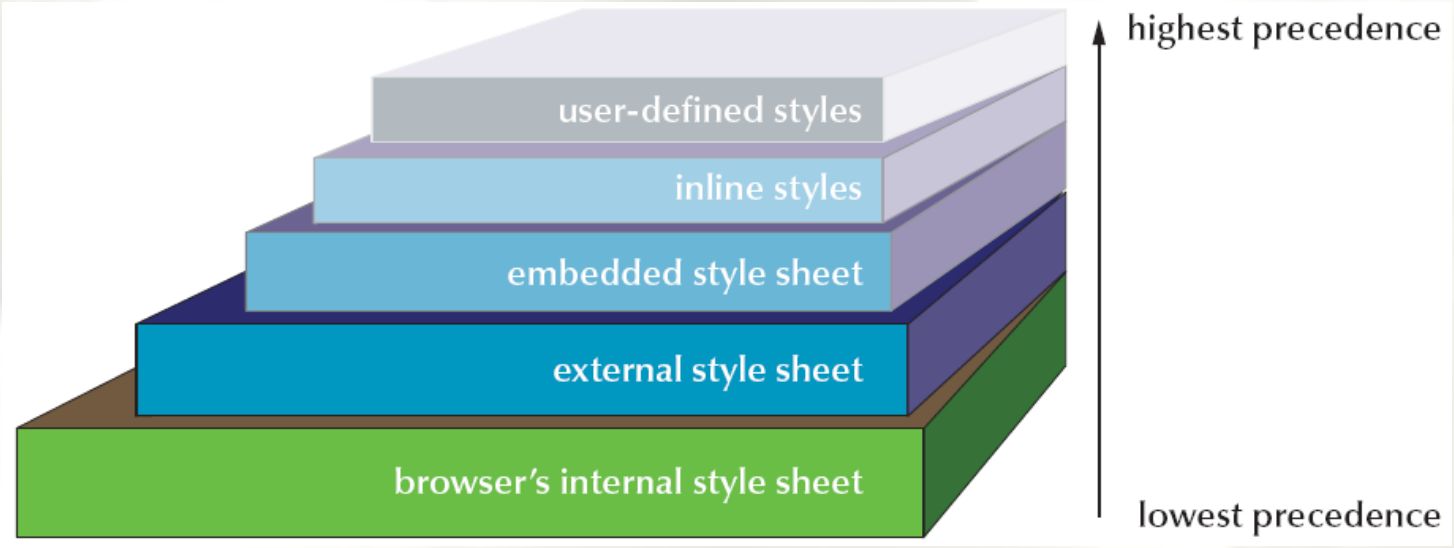
- You can add style comments as you develop an external style sheet

  ```
  /* comment */
  ```

- Use the link element to link a Web page to an external style sheet
- You can import the content of one style sheet into another

Understanding Cascading Order

- You can link a single style sheet to multiple documents in your Web site by using the link element or the @import element
- You can also link a single document to several style sheets



Style Inheritance

- If a style is not specified for an element, it inherits the style of its parent element. This is called **style inheritance**

```
body {color: blue}
p {color: red}
```

- In the above example, the body element is the parent element

Applying a Style to a Specific ID

- To apply a style to an element marked with a specific id value, use the declaration

```
#id {style rule}
```

where id is the value of the element's id attribute and style rule stands for the styles applied to that specific element

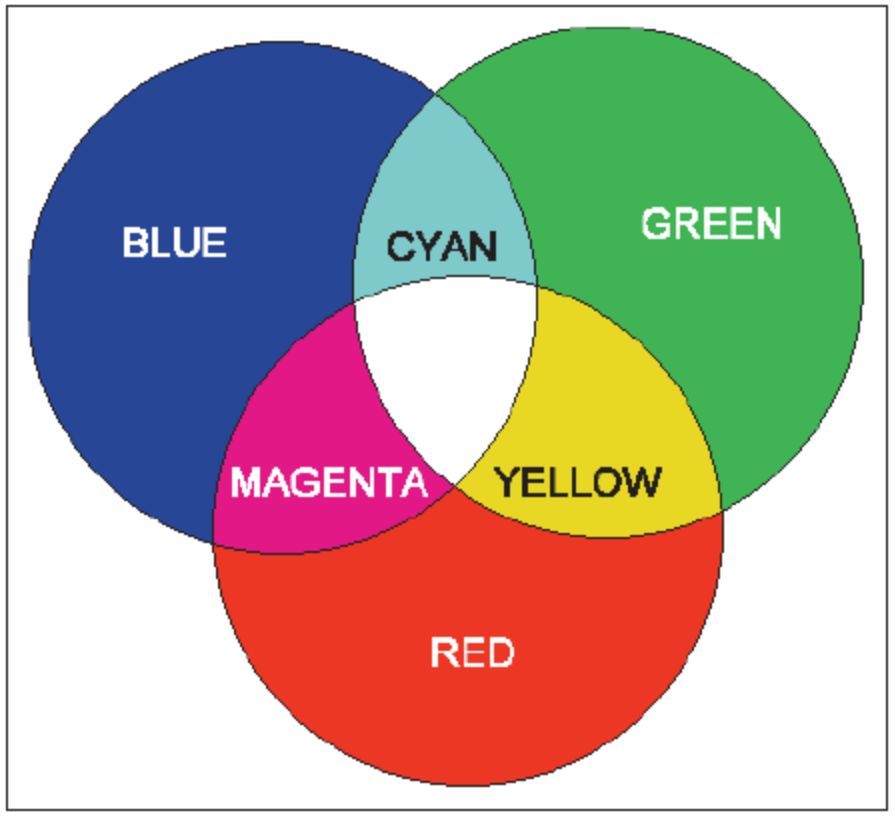
Working with Color in HTML and CSS

- HTML is a text-based language, requiring you to define your colors in textual terms
- HTML identifies a color in one of two ways:
  - By the color value
  - By the color name
- To have more control and more choices, specify colors using **color values**
- A **color value** is a numerical expression that precisely describes a color

Working with Color in HTML and CSS

- Any color can be thought of as a combination of three primary colors: **red**, **green**, and **blue**
- By varying the intensity of each primary color, you can create almost any color and any shade of color
- This principle allows a computer monitor to combine pixels of red, green, and blue to create the array of colors you see on your screen

Working with Color in HTML and CSS



Working with Color in HTML and CSS

- Software programs, such as your Web browser, define color mathematically
- The intensity of each of three colors (RGB) is assigned a number from 0 (absence of color) to 255 (highest intensity)
- In this way, 255^3, or more than 16.7 million, distinct colors can be defined
- Each color is represented by a triplet of numbers, called an **RGB triplet**, based on the strength of its **R**ed, **G**reen, and **B**lue components

Working with Color in HTML and CSS

- HTML requires color values be entered as hexadecimals
- A **hexadecimal** is a number based on a base-16 numbering system rather than a base-10 numbering system that we use every day
  - Base 10 counting uses a combination of 10 characters (0 through 9) to represent numerical values
  - Hexadecimals include six extra characters: A (for 10), B (for 11), C (for 12), D (for 13), E (for 14), and F (for 15)

Working with Color in HTML and CSS

- To represent a number in hexadecimal terms, you convert the value to multiples of 16 plus a remainder. For example:
  - 21 is equal to (16 x 1) + 5, so its hexadecimal representation is 15
  - The number 255 is equal to (16 x 15) + 15, or FF in hexadecimal format (remember that F = 15 in hexadecimal)
  - In the case of the number 255, the first F represents the number of times 16 goes into 255 (which is 15), and the second F represents the remainder of 15
- Once you know the RGB triplet of a color, the color needs to be converted to the hexadecimal format

Working with Color in HTML and CSS

- Using the basic color names allows you to accurately display them across different browsers and operating systems
- The list of only 17 colors is limiting to Web designers

Color Name	RGB Triplet	Hexadecimal	Color Name	RGB Triplet	Hexadecimal
Aqua	(0, 255, 255)	00FFFF	Olive	(128, 128, 0)	808000
Black	(0, 0, 0)	000000	Orange	(255, 165, 0)	FFA500
Blue	(0, 0, 255)	0000FF	Purple	(128, 0, 128)	800080
Fuchsia	(255, 0, 255)	FF00FF	Red	(255, 0, 0)	FF0000
Gray	(128, 128, 128)	808080	Silver	(192, 192, 192)	C0C0C0
Green	(0, 128, 0)	008000	Teal	(0, 128, 128)	008080
Lime	(0, 255, 0)	00FF00	White	(255, 255, 255)	FFFFFF
Maroon	(128, 0, 0)	800000	Yellow	(255, 255, 0)	FFFF00
Navy	(0, 0, 128)	000080			

Working with Color in HTML and CSS

Partial list of extended color names

Sample	Name	RGB	Hexadecimal
	aliceblue	(240,248,255)	#F0F8FF
	antiquewhite	(250,235,215)	#FAEBD7
	aqua	**(0,255,255)**	**#00FFFF**
	aquamarine	(127,255,212)	#7FFFD4
	azure	(240,255,255)	#F0FFFF
	beige	(245,245,220)	#F5F5DC
	bisque	(255,228,196)	#FFE4C4
	black	**(0,0,0)**	**#000000**
	blanchedalmond	(255,235,205)	#FFEBCD
	blue	**(0,0,255)**	**#0000FF**
	blueviolet	(138,43,226)	#8A2BE2
	brown	(165,42,42)	#A52A2A
	burlywood	(222,184,135)	#DEB887

Defining Text and Background Colors

- Background color definition:
  - **`background-color: color`**
- Text color definition:
  - **`color: color`**

  where ***`color`*** is either the color value or the color name
- You can apply text and background colors to any page element

Working with Fonts and Text Styles

- A **specific font** is a font such as Times New Roman, Arial, or Garamond. The font is installed on a user's computer
- A **generic font** refers to the font's general appearance

Generic fonts

	Font Samples		
serif	defg	defg	defg
sans-serif	defg	defg	defg
monospace	defg	defg	defg
cursive	defg	defg	defg
fantasy	defg	defg	DEFG

Working with Fonts and Text Styles

- CSS allows you to specify a list of specific fonts along with a generic font
- If the browser cannot find any of the specific fonts listed, it uses the generic font

```
font-family: Arial, Helvetica,
'Trebuchet MS', sans-serif
```

```
body    {background-color: white; font-family: Arial, Helvetica, sans-serif}
address {text-align: center}
```

Setting the Font Size

- The style to change the font size of text within an element is:
  - **`font-size:` *`length`***

  where ***`length`*** is a length of measure
- **Absolute units** define a font size using one of five standard units of measurement:
  - Millimeters (mm)
  - Centimeters (cm)
  - Inches (in)
  - Points (pt)
  - Picas (pc)

Setting the Font Size

- Many Web page designers opt to use **relative units**, which are expressed relative to the size of other objects within the Web page
  - **Em unit**
  - Percentages
  - Relative keywords
    - Larger
    - Smaller

```
body    {background-color: white; font-family: Arial, Helvetica, sans-serif}
h2      {font-size: 2em}
address {text-align: center}
```

h2 headings will be twice the size of body text

Spacing and Indentation

- **Kerning** is the amount of space between characters
  - **`letter-spacing: value`**
- **Tracking** is the amount of space between words and phrases
  - **`word-spacing: value`**
- **Leading** is the space between lines of text
  - **`line-height: length`**

Applying Font Features

- To specify font styles, use the following style:
  - **font-style: *type***
- To control font weight for any page element, use the following style:
  - **font-weight: *weight***
- To change the appearance of your text, use the following style:
  - **text-decoration: *type***

Applying Font Features

- Underline, overline:
  - **text-decoration: *underline overline***
- Capitalize:
  - **text-transform: *capitalize***
- Uppercase letters, small font:
  - **font-variant: *type***

Aligning Text Vertically

- Use the vertical-align attribute

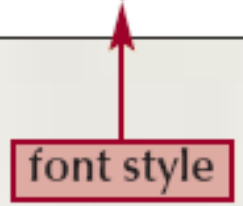
Value	Description
baseline	Aligns the element with the bottom of lowercase letters in surrounding text (the default)
bottom	Aligns the bottom of the element with the bottom of the lowest element in surrounding content
middle	Aligns the middle of the element with the middle of the surrounding content
sub	Subscripts the element
super	Superscripts the element
text-bottom	Aligns the bottom of the element with the bottom of the font of the surrounding content
text-top	Aligns the top of the element with the top of the font of the surrounding content
top	Aligns the top of the element with the top of the tallest object in the surrounding content

Combining All Text Formatting in a Single Style

- You can combine most of them into a single declaration, using the style

```
font: font-style font-variant font-weight font-size/line-height font-family
```

```
body    {background-color: white; font-family: Arial, Helvetica, sans-serif}
h2      {font-size: 2em; letter-spacing: 0.4em; text-indent: 1em }
address {text-align: center; font: normal small-caps 0.8em sans-serif}
```



Working with GIF Images

- **GIF (Graphics Interchange Format)** is the most commonly used image format on the Web
- Compatible with virtually all browsers
- GIF files are limited to displaying 256 colors
- Often used for graphics requiring fewer colors, such as clip art images, line art, logos, and icons
- Images that require more color depth, such as photographs, can appear grainy when saved as GIF files

Working with GIF Images

- A **transparent color** is a color that is not displayed when the image is viewed in an application
- A **splash screen** is a Web page containing interesting animation or graphics that introduces a Web site

JPEG Images

- **JPEG** stands for **Joint Photographic Experts Group**
- Supports up to 16.7 million colors
- Most often used for photographs and other images that cover a wide spectrum of color
- Usually smaller than their GIF counterparts

PNG Images

- A file format called **PNG (Portable Network Graphics)** has been gaining wider acceptance
- PNG files use a free and open file format and can display more colors than GIFs
- PNGs do allow transparent colors, but not all browsers support this feature

Setting the Image Size

- By default, browsers display an image at its saved size
- You can specify a different size by adding the HTML attributes

`width="value" height="value"`

```
<body>
   <div style="text-align: center">
      <img src="salogo.gif" alt="Sunny Acres" width="599" height="223" /><br />
      <a href="home.htm"><img src="scarecrow.gif" alt="animated GIF"
                          width="500" height="300" style="border-width: 0" /></a>
   </div>
</body>
```

Formatting Backgrounds

- The syntax for inserting a background image is: **`background-image: url(`*`url`*`)`**
  - ***URL*** is the location and filename of the graphic file you want to use for the background of the Web page

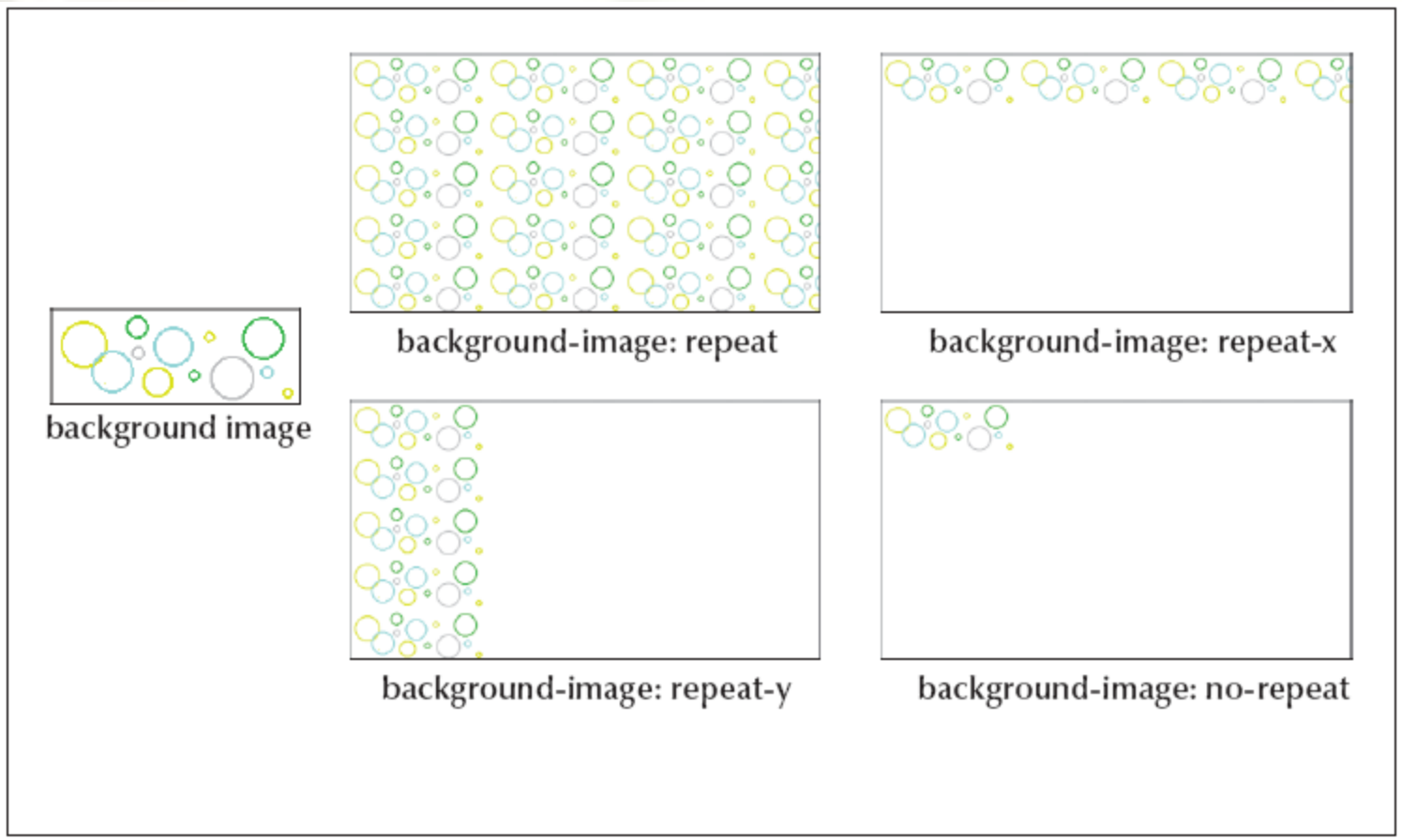
```
<body style="background-image: url(background.jpg)">
   <div style="text-align: center">
      <img src="salogo.gif" alt="Sunny Acres" width="599" height="223" /><br />
      <a href="home.htm"><img src="scarecrow.gif" alt="animated GIF"
                          width="500" height="300" style="border-width: 0" /></a>
   </div>
</body>
```

Background Image Options

- By default, background images are tiled both horizontally and vertically until the entire background of the element is filled up
- You can specify the direction of the tiling using the style:
  - **`background-repeat: type`**

Value	Description
repeat	The image is tiled both horizontally and vertically until the entire background of the element is covered
repeat-x	The image is tiled only horizontally across the width of the element
repeat-y	The image is tiled only vertically across the height of the element
no-repeat	The image is not repeated at all

Background Image Options



The Background Style

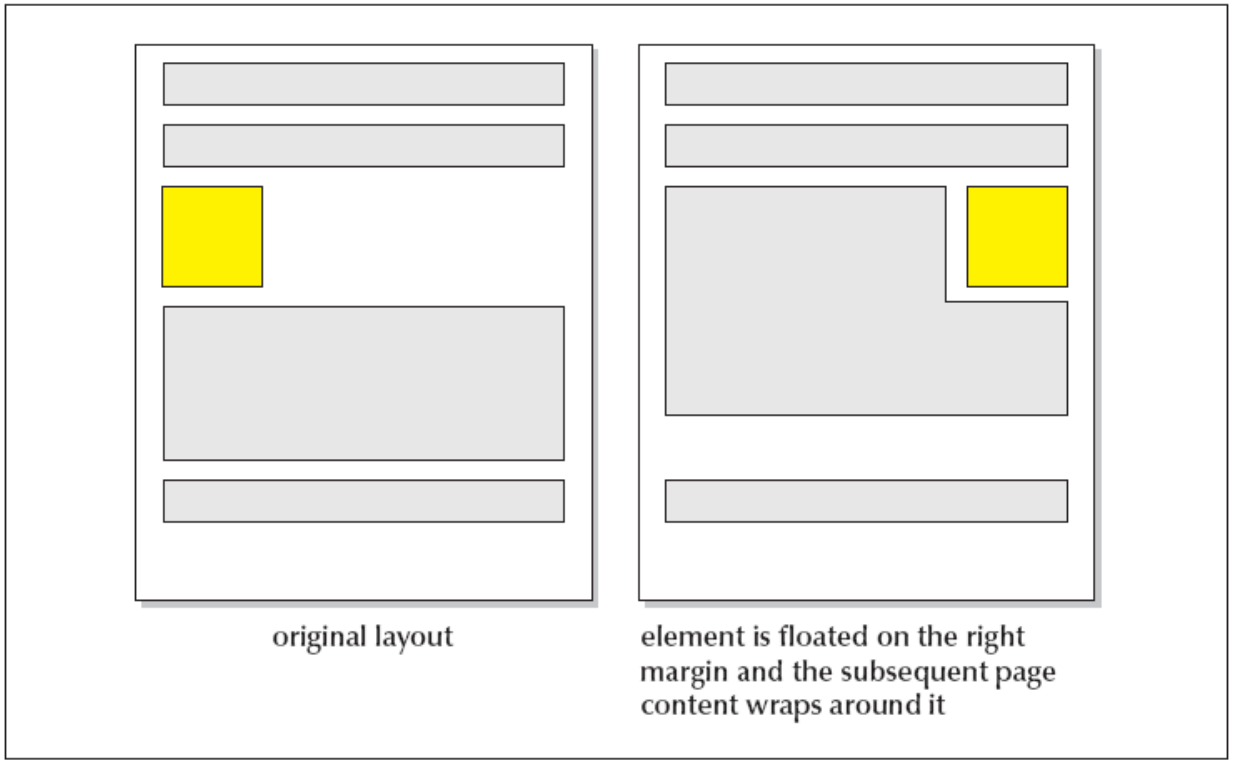
- You can combine the various background styles into the following single style:

```
background: color url(url) repeat attachment horizontal vertical
```

Floating an Element

- The syntax for the float style is:

  `float: position`



Floating an Element

- To float an element, use the style

  `float: position`

  where *position* is none (to turn off floating), left or right
- To display an element clear of a floating element, use the style

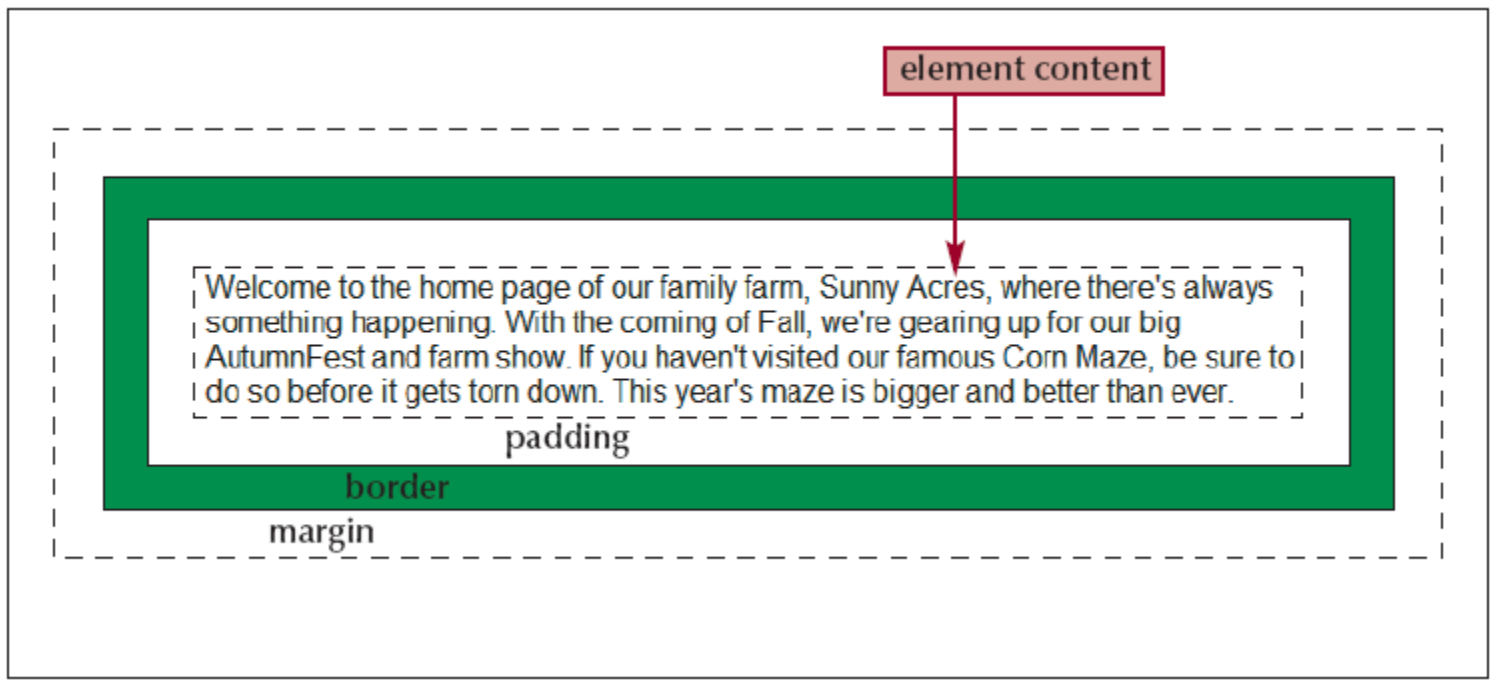
  `clear: position`

  where *position* is none, left, right, or both

Working with the Box Model

- The **box model** describes the structure of page elements as they are laid out on the Web page:
  - The **margin** between the element and other page content
  - The **border** of the box containing the element content
  - The **padding** between the element's content and the box border
  - The **content** of the element itself

Working with the Box Model



Margin Styles

- Control your margins with the following four styles:
  - **margin-top:** ***length***
  - **margin-right:** ***length***
  - **margin-bottom:** ***length***
  - **margin-left:** ***length***
- Margin values can also be negative. This creates an overlay effect by forcing the browser to render one element on top of another
- You can also combine the four margin styles into a single style:
  - ***margin: top right bottom left***

Padding Styles

- Styles to set padding are similar to styles to set margins:
  - **padding-top: *value***
  - **padding-right: *value***
  - **padding-bottom: *value***
  - **padding-left: *value***

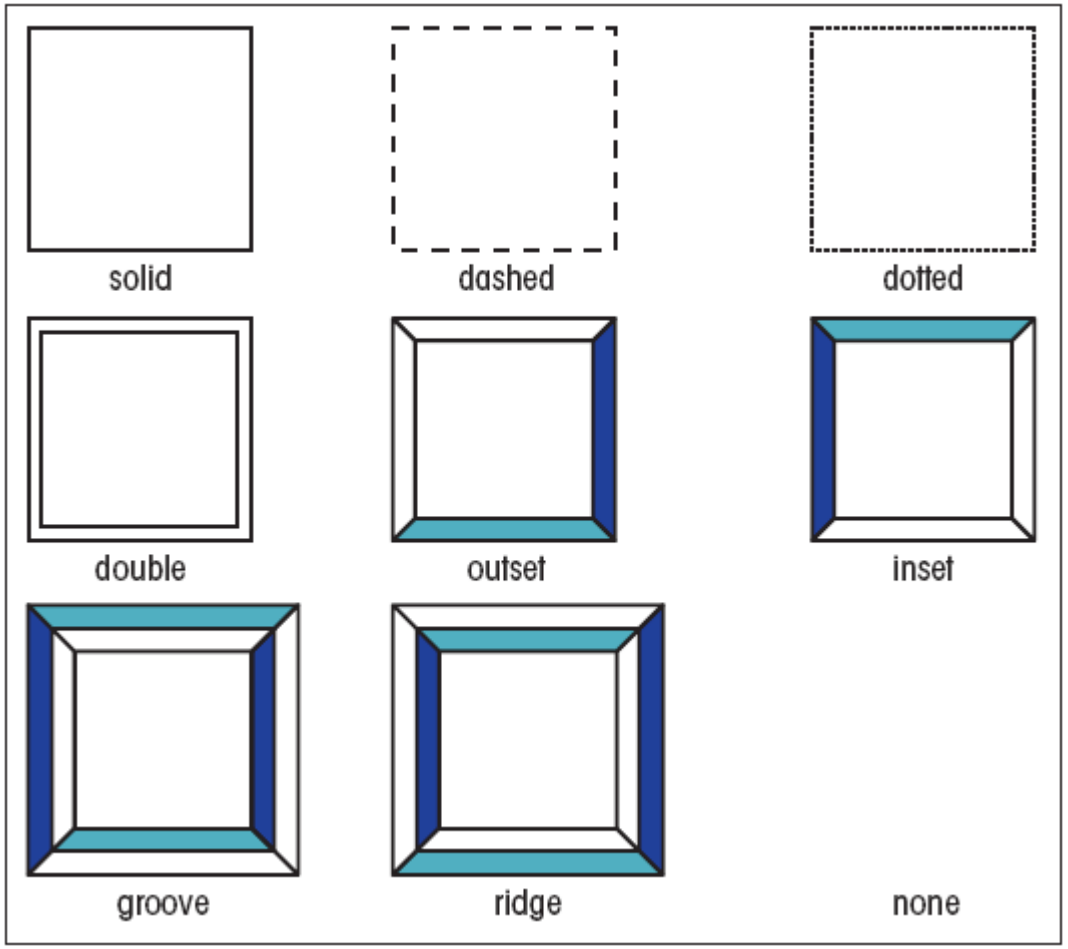
Border Styles

- border-top-width: *length*
- border-right-width: *length*
- border-bottom-width: *length*
- border-left-width: *length*
- border-width: *top right bottom left*
- border-top-color: *color*
- border-right-color: *color*
- border-bottom-color: *color*
- border-left-color: *color*
- border-color: *top right bottom left*

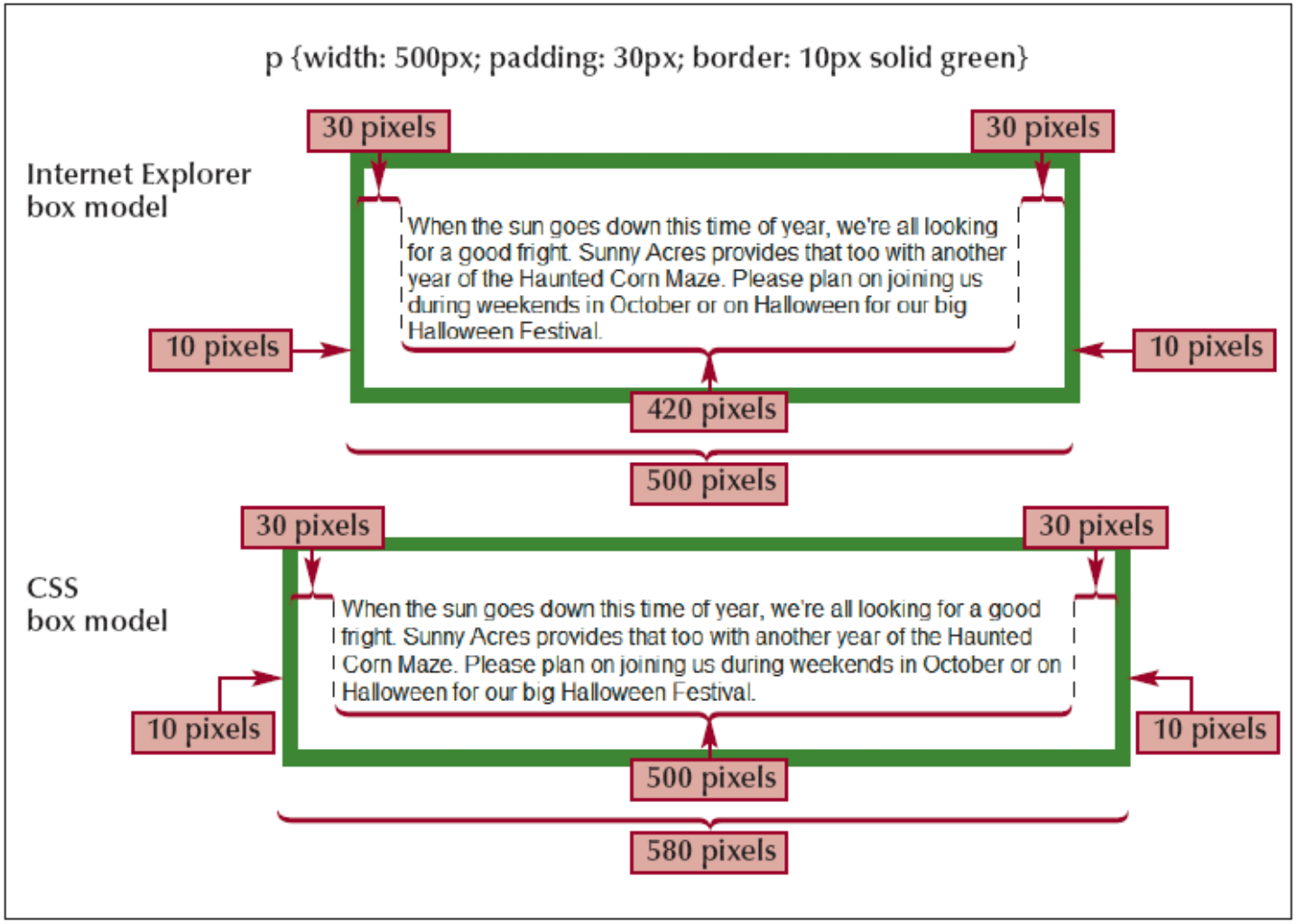
Border Styles

- border-top-style: *type*
- border-right-style: *type*
- border-bottom-style: *type*
- border-left-style: *type*
- border-style: *top right bottom left*

Border Styles



Width and Height Styles



Width and Height Styles

- To set the box model width, use
  - width: *length*

  where *length* is the width of the box content in one of the CSS units of measure. (Note that Internet Explorer applies the width value to the box model content, padding space, and border)
- To set the box model height, use
  - height: *length*

  where *length* is the height of the box content in one of the CSS units of measure

Controlling Page Layout with div Containers

- **div containers** can be resized and floated to create different page layouts

```
<body>

<div id="outer">
   <h1><img src="salogo.jpg" alt="Sunny Acres" /></h1>
   <div id="links">
      <a href="home.htm">Home</a>
      <a href="maze.htm">The Corn Maze</a>
      <a href="haunted.htm">The Haunted Maze</a>
      <a href="petting.htm">Petting Barn</a>
      <a href="produce.htm">Produce</a>
   </div>

   <h2>Welcome</h2>

   <address>
      Sunny Acres  &#9728; 
      Tammy &amp; Brent Nielsen  &#9728; 
      1977 Highway G  &#9728; 
      Council Bluffs, IA    51503
   </address>

</div>
</body>
```

Controlling Page Layout with div Containers

```
body    {background-color: white; font-family: Arial, Helvetica, sans-serif}
h2      {font-size: 2em; letter-spacing: 0.4em; text-indent: 1em }
h3      {width: 20em; padding-left: 1em}
address {text-align: center; font: normal small-caps 0.8em sans-serif;
         border-top: 0.5em double green; padding-top: 1em}

#promoimage {float: right; margin: 0em 0em 1em 1em}
#outer      {width: 50em}
```

```
#promoimage {float: right; margin: 0em 0em 1em 1em}
#outer      {width: 50em}
#links      {float: left; width: 10em; background-color: white;
             border-style: outset; border-width: 0.5em}
```

Setting the Display Style

Values of the display style

Display	Description
block	Display as a block-level element
inline	Display as an inline element
inline-block	Display as an inline element with some of the properties of a block (much like an inline image or frame)
inherit	Inherit the display property of the element's parent
list-item	Display as a list item
none	Do not display the element
run-in	Display as either an inline or block-level element depending on the context (CSS2)
table	Display as a block-level table
inline-table	Display as an inline table
table-caption	Treat as a table caption
table-cell	Treat as a table cell
table-column	Treat as a table column
table-column-group	Treat as a group of table columns
table-footer-group	Treat as a group of table footer rows
table-header-group	Treat as a group of table header rows
table-row	Treat as a table row
table-row-group	Treat as a group of table rows

Setting the Display Style

- To set the display style of an element, use
  - display: *type*

  where *type* is the type of display. Use inline for inline elements and block for block-level elements

Summary

- Learned history and concepts of CSS
- Learned different styles and how they are applied
- Learned CSS use of color and CSS styles for font
- Learned to display an animated graphic
- Learned to float elements and apply style to elements
- Learned the properties of the box model