

Tutorial 3

Designing a Web Page with CSS

HTML, CSS, and Dynamic HTML

5TH EDITION



Objectives

- Explore the history and theory of CSS
- Define a style rule
- Study style precedence and inheritance
- Apply color using CSS
- Explore CSS3 color extensions
- Use contextual selectors

Objectives

- Work with attribute selectors
- Apply text and font styles
- Install a Web font
- Define list styles
- Use pseudo-classes and pseudo-elements
- Create a rollover effect

Style Sheets and Color

Style comments provide information about the style sheet.

```
/* Sunny Acres Style Sheet
   Author: Tammy Nielsen
   Date: 3/1/2014
*/

/* Body styles */
body {
  background-color: white;
  font-family: Verdana, Geneva, sans-serif;
  line-height: 1.4em;
}
```

The appearance of the Web page is determined by the styles in a style sheet.

Every style rule needs to be enclosed in curly braces, with the style property values separated by semicolons.

The **selector** defines what element or elements are affected by a rule.

```
h1 {
  color: white;
  background-color: rgb(50, 69, 99);
}
```

The background-color style property sets the background color.

The color style property sets the text color.

CSS supports **color names** for a select group of commonly used colors.

The **HSL model** selects color from a color wheel at varying levels of saturation and lightness.

h2 {
 background-color: rgb(0, 165, 0);
 color: rgba(255, 255, 255, 0.8);
}

The color is 80% opaque.

Color values using the rgba or hsla properties can include opacity to create semi-transparent colors.

Sunny Acres

Tammy and Brent Nielsen
1973 Hwy G
Council Bluffs, IA 51503

HOME

Autumn Fun

Scary Good

Meet the Animals

For your Tastebuds

The Farm Shop

The Sunny Acres Farm Shop aims to offer the highest quality fresh produce. You can pick your own or buy it in our shop. Set amidst acres of outstanding natural beauty on the beautiful rolling hills northeast of Council Bluffs, the Farm Shop is easily reached on Highway G, with easy access from Interstate 80.

The Farm Shop was established over 20 years ago with great success. Our products have won numerous awards at local festivals and fairs. We also cater to local supermarkets in the Council Bluffs/Omaha area. Look for our products every Saturday morning from May to October at the Council Bluffs Farmers' Market.

Hours

- Monday - Friday: 9 am - 5 pm
- Saturday: 9 am - 3 pm
- Pick Your Own Produce is available from May 15 - October 22
- The Farm Shop is open year-round

Products

- Freshly baked breads and quiches
- High quality meats
- Cheese and other dairy products
- Freshly-picked fruits and vegetables (in season)
- Canned goods and preserves

SUNNY ACRES * TAMMY & BRENT NIELSEN * 1977 HIGHWAY G * COUNCIL BLUFFS, IA 51503

CSS Color Models

rgb(red, green, blue)

red, green, and blue range from 0 (no intensity) up to 255 (highest intensity)

hsl(hue, saturation, lightness)

hue ranges from 0 to 360 degrees, saturation and lightness vary from 0% to 100%

pink
brown
orange
seagreen
powderblue

The **RGB model** combines red, blue, and green color of varying intensities.

Introducing CSS

- To render a document, the device displaying the page needs a style sheet that specifies the appearance of each page element
- The style sheet language used on the Web is the Cascading Style Sheets language, also known as CSS
 - Versions include **CSS1**, **CSS2**, **CSS 2.1**, and **CSS3**
- With CSS, as with HTML, Web page designers need to be aware of compatibility issues that arise not just among different versions of the language, but also among different versions of the same browser

Defining a Style Rule

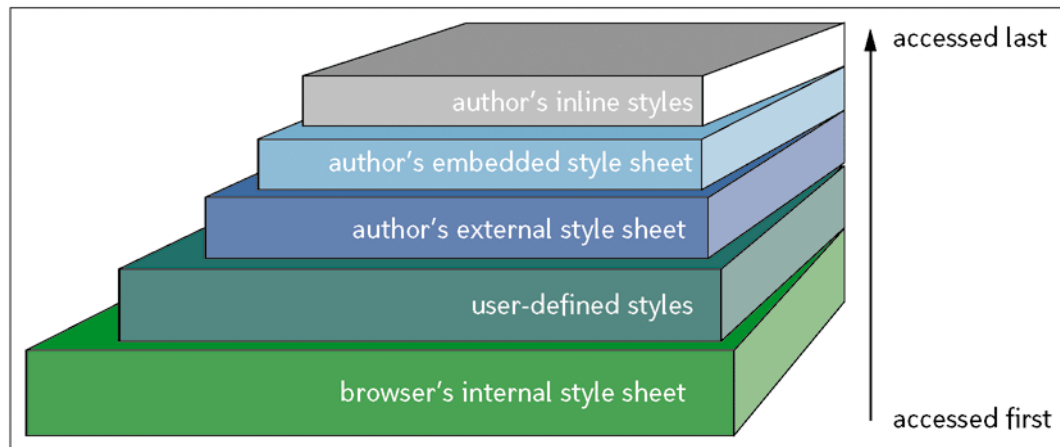
- The general syntax of a CSS style rule is

```
selector {  
    property1: value1;  
    property2: value2;  
    property3: value3;  
  
    . . .  
}
```

Applying a Style Sheet

- The design you apply to a Web site is usually a combination of several style sheets

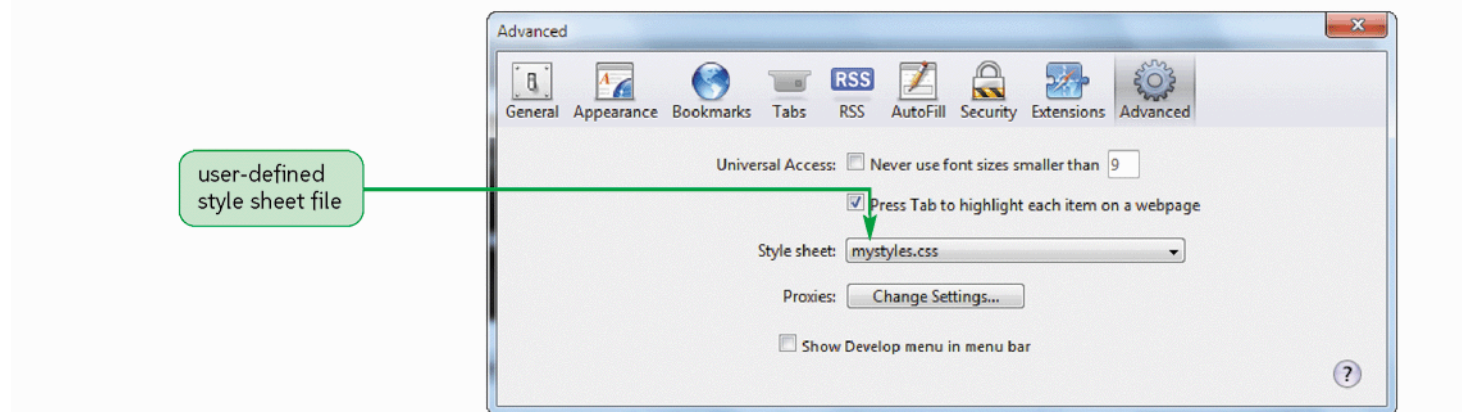
Figure 3-4 Order in which style sheets are interpreted



User-Defined Styles

- Almost all browsers allow users to modify the default settings of the internal style sheet

Figure 3-5 Choosing a user-defined style sheet in Safari



External Style Sheets

Figure 3-6 Linking to an external style sheet

style sheet file

```
<meta charset="UTF-8" />  
<title>Sunny Acres</title>  
<script src="modernizr-1.5.js"></script>  
<link href="sa_layout.css" rel="stylesheet" type="text/css" />  
</head>
```

Embedded Style Sheets

- Another type of style sheet created by a Web page author is an embedded style sheet, in which the styles are inserted directly within the head element of an HTML document using the style element

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

```
    styles
```

```
</style>
```

- The exact order in which external style sheets and embedded style sheets are processed by the browser depends on the order in which they are listed within the HTML file

Inline Styles

- The very last styles to be interpreted by the browser are inline styles, which are styles applied directly to specific elements using the style attribute

```
<element style="style rules"> ...  
</element>
```

- It is clear exactly what page element is being formatted
- Not recommended in most cases and considered inefficient

Exploring the Style Cascade

- As a general rule of thumb, all other things being equal, the more specific style is applied instead of the more general
- An additional factor in applying a style sheet is that properties are passed from a parent element to its children in a process known as **style inheritance**

```
body {color: blue;}
```

```
h1 {text-align: center;}
```

Exploring the Style Cascade

- If you need browsers to enforce a style, you can append the `!important` keyword to the style property, using the syntax

```
property: value !important;
```
- The `!important` keyword is often necessary for visually impaired users who require their pages rendered with large, clear text and highly contrasting colors

Writing Style Comments

Figure 3-8 Entering style sheet comments

```
/*  
    Sunny Acres Style Sheet  
    Author: Tammy Nielsen  
    Date:   3/1/2014  
*/
```

Defining Color in CSS

- A **color value** is a numerical expression that describes the properties of a color
- CSS represents these intensities mathematically as a set of numbers called an **RGB triplet**, which has the format

`rgb(red, green, blue)`

- CSS also allows RGB values to be entered as **hexadecimal** numbers

`#redgreenblue`

Defining Color in CSS

Figure 3-10 The 16 basic CSS2 color names

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

Setting Foreground and Background Color

- To set the background color of an element, use the property

`background-color: color;`

where *color* is a color name or a color value.

- To set the foreground or text color of an element, use the following property:

`color: color;`

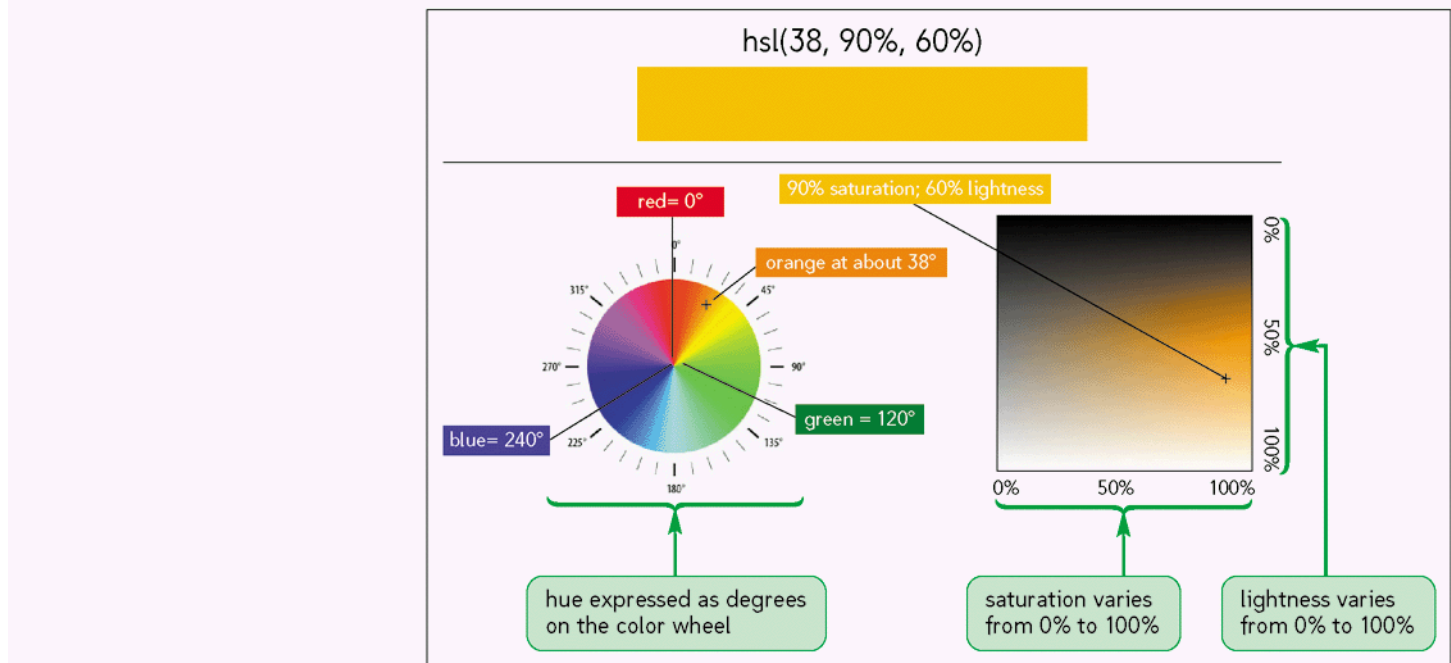
Enhancements to Color in CSS3

- CSS3 also supports the Hue Saturation Lightness (HSL) model that describes colors based on **hue**, **saturation**, and **lightness**

`hsl(hue, saturation,
lightness)`

Enhancements to Color in CSS3

Figure 3-14 HSL color saturation model



Enhancements to Color in CSS3

- CSS3 also allows page designers to augment RGB and HSL color values by specifying a color's opacity. Opacity defines how much of the colors below the surface of the current object show through to affect its appearance

`rgba(red, green, blue, opacity)`

`hsla(hue, saturation, lightness,
opacity)`

Enhancements to Color in CSS3

Figure 3-15 Setting a semi-transparent color

older browsers
will display h2
text in white

```
h2 {  
  background-color: rgb(0, 165, 0);  
  color: white;  
  color: rgba(255, 255, 255, 0.8);  
}
```

white color with
80% opacity

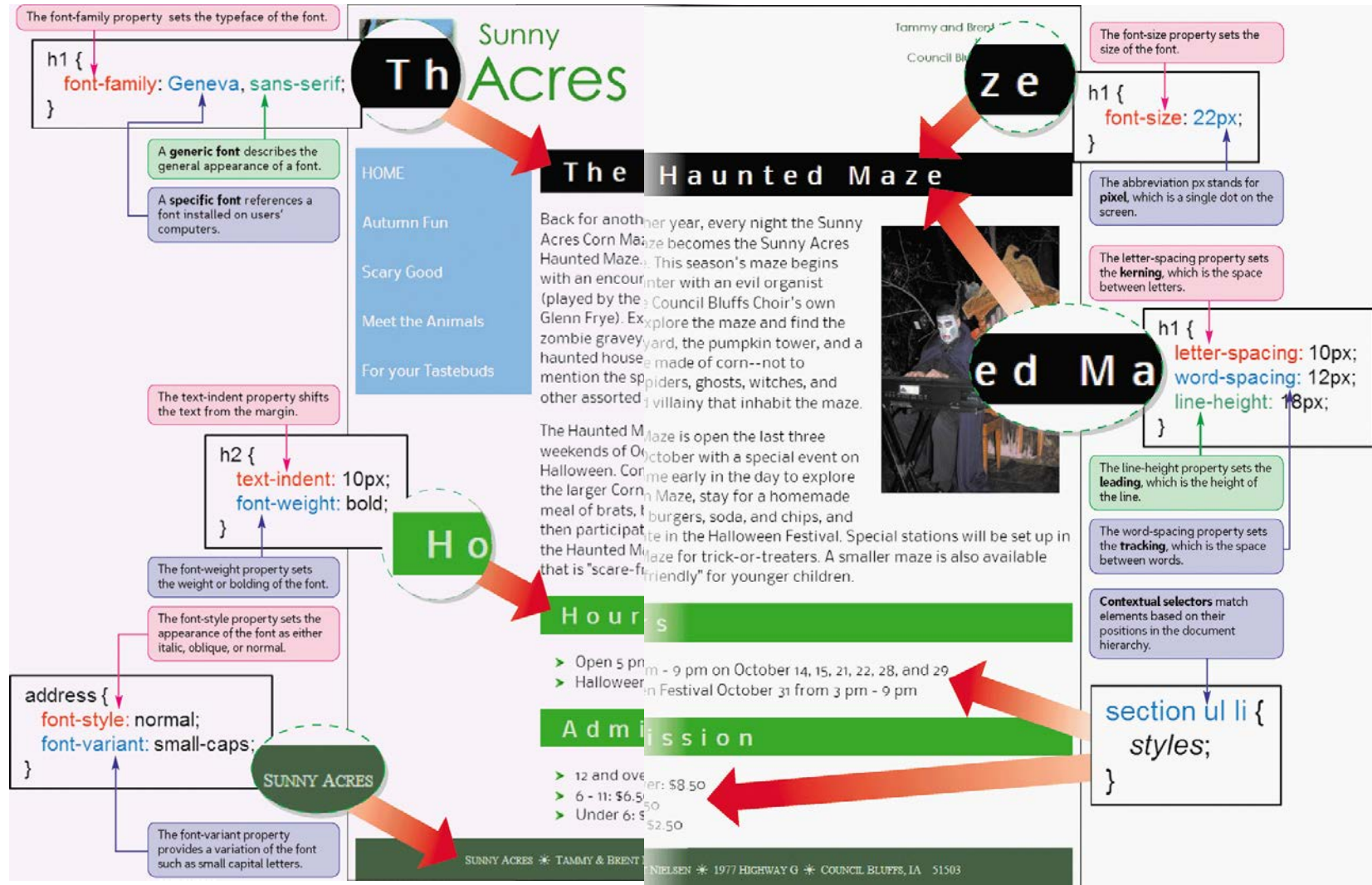
Figure 3-16 Heading text in semi-transparent white

green background bleeds
through the semi-transparent
white text

Hours

- Farm Shop: 9 am - 5 pm Mon - Fri; 9 am - 3 pm Sat
- The Corn Maze: 11 am - 9 pm Sat; 11 am - 5 pm Sun
- The Haunted Maze: 5 pm - 9 pm Fri & Sat
- Petting Barn: 9 am - 4 pm Mon - Fri; 11 am - 3 pm Sat & Sun

Selectors and Text Styles



Contextual Selectors

- Web pages are structured documents in which elements are nested within other elements, forming a hierarchy of elements
- To create styles that take advantage of this tree structure, CSS allows you to create contextual selectors whose values represent the locations of elements within the hierarchy
 - **Parent elements**
 - **Child elements**
 - **Sibling elements**
 - **Descendant elements**

Contextual Selectors

Figure 3-20 Contextual selectors

Selector	Description
*	Matches any element in the hierarchy
<i>e</i>	Matches any element, <i>e</i> , in the hierarchy
<i>e1, e2, e3, ...</i>	Matches the group of elements <i>e1, e2, e3, ...</i>
<i>e f</i>	Matches any element, <i>f</i> , that is a descendant of an element, <i>e</i>
<i>e>f</i>	Matches any element, <i>f</i> , that is a direct child of an element, <i>e</i>
<i>e+f</i>	Matches any element, <i>f</i> , that is immediately preceded by a sibling element, <i>e</i>
<i>e~f</i>	Matches any element, <i>f</i> , that is a sibling to an element, <i>e</i>

Figure 3-22 Applying a contextual selector

style rule applied only to h1 headings
nested within section elements

```
section h1 {  
  background-color: rgb(125, 186, 240);  
  color: white;  
}
```


Attribute Selectors

- Selectors also can be defined based on attributes and attribute values associated with elements
- Two attributes, `id` and `class`, are often key in targeting styles to a specific element or group of elements

Attribute Selectors

Figure 3-23 Attribute selectors

Selector	Description	Example	Matches
<code>#id</code>	The element with the id value, <i>id</i>	<code>#intro</code>	The element with the id <i>intro</i>
<code>.class</code>	All elements with the class value, <i>class</i>	<code>.main</code>	All elements belonging to the <i>main</i> class
<code>elem.class</code>	All <i>elem</i> elements with the class value <i>class</i>	<code>p.main</code>	All paragraphs belonging to the <i>main</i> class
<code>elem[att]</code>	All <i>elem</i> elements containing the <i>att</i> attribute	<code>a[href]</code>	All hypertext elements containing the href attribute
<code>elem[att="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute equals <i>text</i>	<code>a[href="gloss.htm"]</code>	All hypertext elements whose href attribute equals <i>gloss.htm</i>
<code>elem[att~="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute contains the word <i>text</i>	<code>a[rel~="glossary"]</code>	All hypertext elements whose rel attribute contains the word <i>glossary</i>
<code>elem[att ="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute value is a hyphen-separated list of words beginning with <i>text</i>	<code>p[id "first"]</code>	All paragraphs whose id attribute starts with the word <i>first</i> in a hyphen-separated list of words
<code>elem[att^="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute begins with <i>text</i> (CSS3)	<code>a[rel^="prev"]</code>	All hypertext elements whose rel attribute begins with <i>prev</i>
<code>elem[att\$="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute ends with <i>text</i> (CSS3)	<code>a[href\$="org"]</code>	All hypertext elements whose href attribute ends with <i>org</i>
<code>elem[att*="text"]</code>	All <i>elem</i> elements whose <i>att</i> attribute contains the value <i>text</i> (CSS3)	<code>a[href*="faq"]</code>	All hypertext elements whose href attribute contains the text string <i>faq</i>

Styling Web Page Text

- The default font used by most browsers is Times New Roman, but you can specify a different font for any page element using the property

`font-family: fonts;`

- `fonts` is a comma-separated list of specific or generic font names

Styling Web Page Text

Figure 3-26

Web safe fonts

Arial

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: Arial, Helvetica, sans-serif;

Arial Black

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Arial Black', Gadget, sans-serif;

Century Gothic

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Century Gothic', sans-serif;

Comic Sans MS

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Comic Sans MS', cursive;

Courier New

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Courier New', Courier, monospace;

Georgia

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: Georgia, serif;

Impact

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: Impact, Charcoal, sans-serif;

Lucida Console

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Lucida Console', Monaco, monospace;

Lucida Sans Unicode

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Lucida Sans Unicode', 'Lucida Grande', sans-serif;

Palatino Linotype

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Palatino Linotype', 'Book Antiqua', Palatino, serif;

Tahoma

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: Tahoma, Geneva, sans-serif;

Times New Roman

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Times New Roman', Times, serif;

Trebuchet MS

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: 'Trebuchet MS', Helvetica, sans-serif;

Verdana

abcdefghijklmnopqrstuvwxyz/1234567890
font-family: Verdana, Geneva, sans-serif;

Setting Font Face and Sizes

- To define a font face, use the style property

`font-family: fonts;`

where *fonts* is a comma-separated list of fonts that the browser can use with the element. List specific fonts first and complete the list with a generic font.

- To set a font size, use the style property

`font-size: size;`

where *size* is a CSS unit of length in either relative or absolute units.

Setting Font Face and Sizes

- To set kerning (the space between letters), use the following style property:

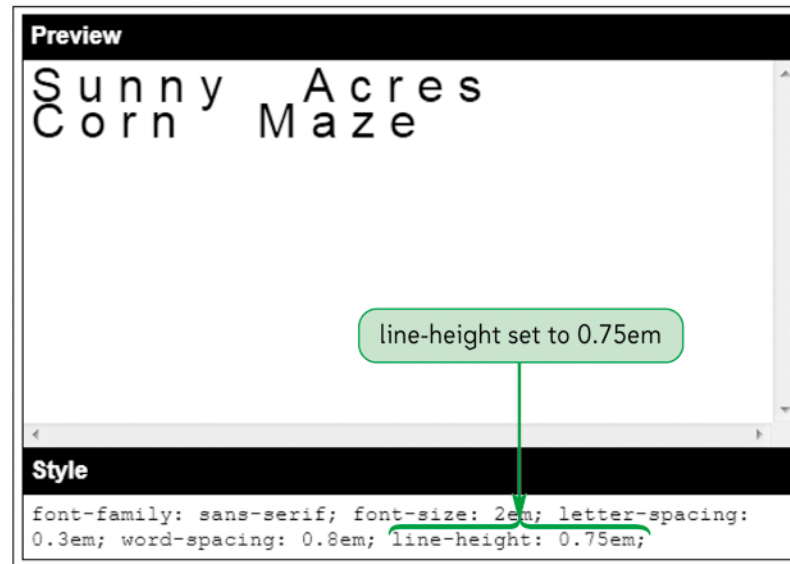
`letter-spacing: size;`

- To set tracking (the space between words), use the following style property:

`word-spacing: size;`

Setting the Line Height

Figure 3-31 Setting the line height



Setting Font and Text Appearance

- To specify the font style, use

`font-style: type;`

where *type* is `normal`, `italic`, or `oblique`.

- To specify the font weight, use

`font-weight: type;`

where *type* is `normal`, `bold`, `bolder`, `light`, `lighter`, or a font weight value.

Difference between italic & oblique

- Italic text is in some way its own font, with small changes made to the structure of each letter to account for the altered appearance.
- Oblique text, on the other hand, is simply a slanted version of the normal, upright text.

italic text sample

oblique text sample

Setting Font and Text Appearance

- To specify a text decoration, use

`text-decoration: type;`

where *type* is none, underline, overline, or line-through.

- To transform text, use

`text-transform: type;`

where *type* is capitalize, uppercase, lowercase, or none.

- To display a font variant of text, use

`font-variant: type;`

where *type* is normal or small-caps.

Aligning Text Vertically

Figure 3-35 Values of the vertical-align style

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 the text and font styles into a single property using the shortcut font property
- `font: font-style font-variant font-weight font-size/line-height font-family;`
- Only font-size & font-family are required.

Combining All Text Formatting in a Single Style

Figure 3-36 Designing the footer address

set the background color to dark and the text color to white or semi-transparent white

shortcut font property

center the text horizontally

```
/* Footer styles */  
footer address {  
  background-color: rgb(55, 102, 55);  
  color: white;  
  color: rgba(255, 255, 255, 0.8);  
  font: normal small-caps 0.8em/4em 'Times New Roman', Times, serif;  
  text-align: center;  
}
```

Figure 3-37 Reformatted address text

address text

Directions

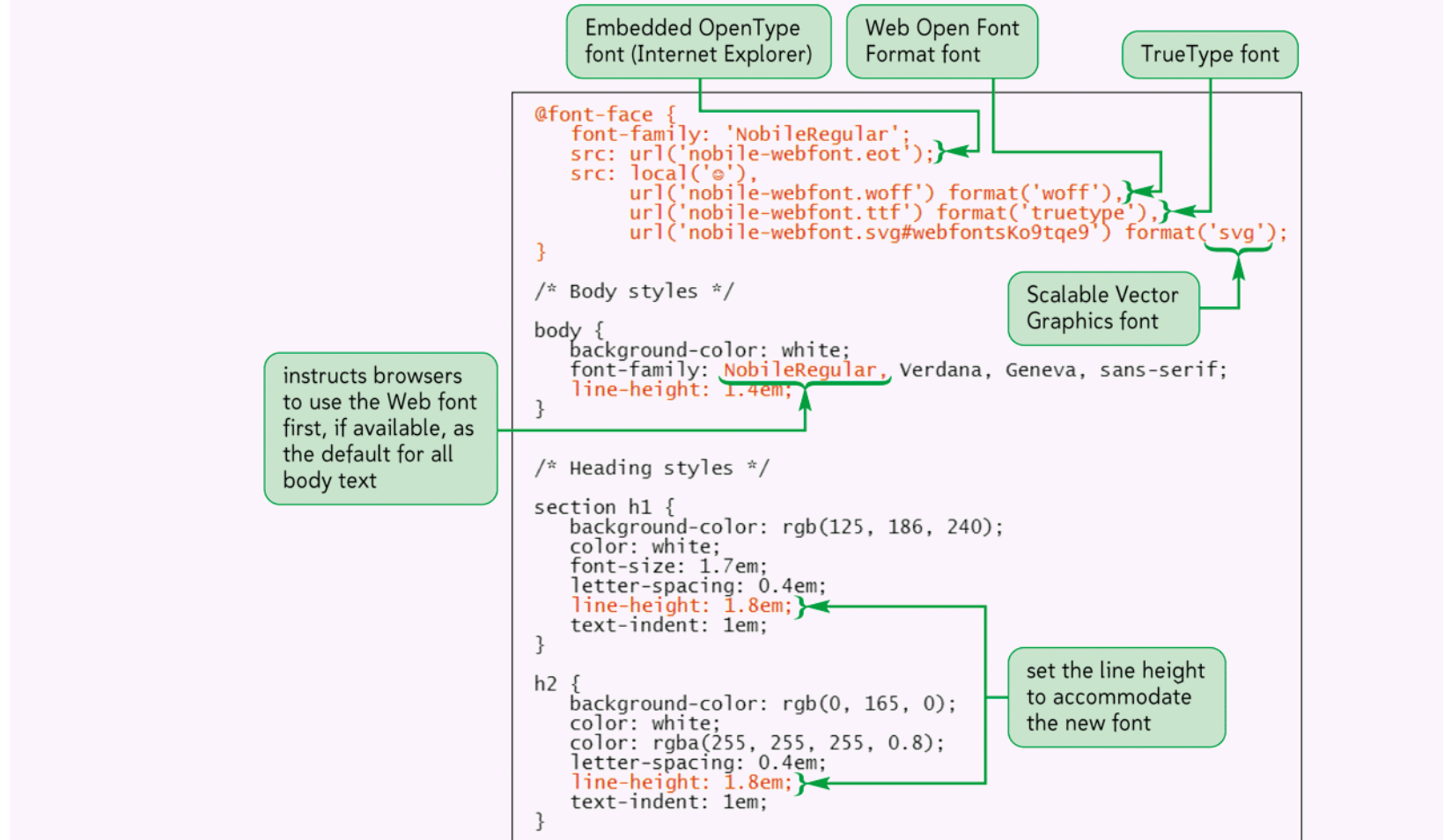
- From Council Bluffs, proceed east on I-80
- Take Exit 38 North to the Drake Frontage Road
- Turn right on Highway G
- Proceed east for 2.5 miles
- Sunny Acres is on your left; watch for the green sign

SUNNY ACRES * TAMMY & BRENT NIELSEN * 1977 HIGHWAY G * COUNCIL BLUFFS, IA 51503

Working with Web Fonts

Figure 3-39

Inserting a Web font



Understanding the CSS @rules

Figure 3-41

CSS @rules

@rule	Description
@charset " <i>encoding</i> ";	Defines the character encoding used in an external style sheet where <i>encoding</i> is the name of the character set
@import url(<i>url</i>) <i>media</i>	Imports an external style sheet file located at <i>url</i> . The optional <i>media</i> attribute provides a comma-separated list of media devices to be used with the style sheet
@media <i>media</i> { <i>styles</i> }	Targets the style rules in <i>styles</i> to devices that match the media types specified in <i>media</i>
@page <i>location</i> { <i>margins</i> }	Defines the page margins for printed output where <i>location</i> is either left, right, or first for left page, right page, or first page, and <i>margins</i> set the margin widths
@font-face { <i>font_description</i> }	Defines the properties of a custom Web font where <i>font_description</i> indicates the source and features of the font
@namespace <i>prefix</i> <i>uri</i>	Defines an XML namespace where <i>prefix</i> is the namespace prefix and <i>uri</i> is the location of the namespace

Lists and Pseudo-Items

The first-of-type structural pseudo-class matches the element in a collection.

```
nav ul li:first-of-type { text-transform: uppercase; }
```

The text-transform property transforms the text in uppercase, lowercase, or sentence case.

The hover dynamic pseudo-class matches the situation when a user is hovering the mouse pointer over the element.

```
nav ul li:hover { background-color: darkblue; }
```

The list-style-type property specifies the marker that is displayed with the list.

```
ul { list-style-type: square; }
```

HOME

Autumn Fun

Scary Good

Meet the Animals

For your Tastebuds

There's Always Something Happening

With the coming of fall, we're gearing up for our big AutumnFest and you haven't visited our famous Corn Maze, be sure to do so before it gets torn down on November 5. This year's maze is bigger and better than ever.

Farms can be educational and Sunny Acres is no exception. Schools and home-schooling parents, spend an afternoon with us at our Petting Barn. We have over 100 friendly farm animals in a clean environment. Kids can bottle feed the baby goats, lambs, and calves while they learn about nature and farming life. Please call ahead for large school groups.

Hours

- Farm Shop: 9 am - 5 pm Mon - Fri; 9 am - 3 pm Sat & Sun
- The Corn Maze: 11 am - 9 pm Mon - Fri; 9 am - 3 pm Sat & Sun
- The Haunted Maze: 5 pm - 9 pm Mon - Fri; 9 am - 4 pm Mon - Fri; 11 am - 3 pm Sat & Sun
- Petting Barn: 9 am - 5 pm Mon - Fri; 9 am - 3 pm Sat & Sun

Directions

- From Council Bluffs, proceed east on I-80
- Take Exit 38 North to the Drake Frontage Road
- Turn right on Highway G
- Proceed east for 2.5 miles
- Sunny Acres is on your left; watch for the green sign

SUNNY ACRES * TAMMY & BRENT NIELSEN * 1977 EDGEMONT RD * COUNCIL BLUFFS, IA 51503

THE FIRST LETTER

```
p:first-letter { font-size: 200%; }
```

The first-letter pseudo-element matches the first letter of the element.

An initial cap effect occurs when the first letter of the first line appears larger than the surrounding text.

The first-line pseudo-element matches the first line of the element.

```
p:first-line { text-transform: uppercase; }
```

OPENING

fall, Fest and

From Council Bluffs

Take Exit 38

Turn right

```
ul { list-style-image: url(arrow.png); }
```

The list-style-image property specifies an image to be displayed as the marker.

Designing Styles for Lists

- To change the marker displayed in ordered or unordered lists, you apply the style

`list-style-type: type;`

Figure 3-42 List style types

list-style-type	Marker (s)
disc	●
circle	○
square	■
decimal	1, 2, 3, 4, ...
decimal-leading-zero	01, 02, 03, 04, ...
lower-roman	i, ii, iii, iv, ...
upper-roman	I, II, III, IV, ...
lower-alpha	a, b, c, d, ...
upper-alpha	A, B, C, D, ...
lower-greek	α, β, γ, δ, ...
upper-greek	Α, Β, Γ, Δ, ...
none	no marker displayed

Designing a List

- To define the appearance of the list marker, use the style

`list-style-type: type;`

where *type* is `disc`, `circle`, `square`, `decimal`, `decimal-leading-zero`, `lower-roman`, `upper-roman`, `lower-alpha`, `upper-alpha`, `lower-greek`, `upper-greek`, or `none`.

- To insert a graphic image as a list marker, use the style

`list-style-image: url(url);`

where *url* is the URL of the graphic image file.

Designing a List

- To set the position of list markers, use the style

`list-style-position: position;`

where *position* is `inside` or `outside` (default).

- To define all of the list style properties in a single style, use the following style:

`list-style: type url(url) position;`

- To set the indentation of a list, apply the style

`padding-left: size;`

where *size* is the length that the list should be indented.

Using Pseudo-Classes and Pseudo-Elements

- A pseudo-class is a classification of an element based on its current status, position, or use in the document

selector:pseudo-class {styles;}

Figure 3-51

Dynamic pseudo-classes

Pseudo-Class	Description	Example
link	The link has not yet been visited by the user.	a:link {color: red;}
visited	The link has been visited by the user.	a:visited {color: green;}
active	The element is in the process of being activated or clicked by the user.	a:active {color: yellow;}
hover	The mouse pointer is hovering over the element.	a:hover {color: blue;}
focus	The element has received the focus of the keyboard or mouse pointer.	input:focus {background-color: yellow;}

Using Pseudo-Classes and Pseudo-Elements

Figure 3-54 Structural pseudo-classes

Pseudo-Class	Matches
root	The top element in the document hierarchy (the html element)
empty	An element with no children
only-child	An element with no siblings
first-child	The first child of the parent element
last-child	The last child of the parent element
first-of-type	The first element of the parent that matches the specified type
last-of-type	The last element of the parent that matches the specified type
nth-of-type(<i>n</i>)	The <i>n</i> th element of the parent of the specified type
nth-last-of-type(<i>n</i>)	The <i>n</i> th from the last element of the parent of the specified type
only-of-type	An element that has no siblings of the same type
lang(<i>code</i>)	The element that has the specified language indicated by <i>code</i>
not(<i>s</i>)	An element not matching the specified selector, <i>s</i>

Pseudo-Elements

Figure 3-56 Pseudo-elements

Pseudo-Element	Description	Example
first-letter	The first letter of the element text	p:first-letter {font-size:200%}
first-line	The first line of the element text	p:first-line {text-transform: uppercase}
before	Content inserted directly before the element	p:before {content:"Special!"}
after	Content inserted directly after the element	p:after {content:"eof"}