Chapter 4
Introduction to Cascading Style Sheets (CSS): Part 1

INTERNET & WORLD WIDE WEB
HOW TO PROGRAM, 5/E

OBJECTIVES

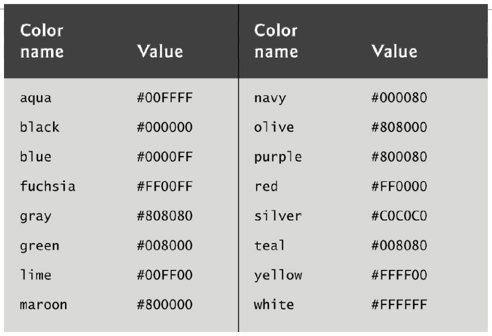
- Control a website's appearance with style sheets
- Use a style sheet to give all the pages of a website the same look and feel
- Use the class attribute to apply style sheets
- Specify the precise font size, color and other properties of displayed text
- Specify element backgrounds and colors.
- Understand the box model and how to control margiris. borders and padding
- Use style sheets to separate presentation from content.

Introduction

- Cascading Style Sheets 3 (CSS3)
  - Used to specify the presentation of elements separately from the structure of the document.
- CSS validator `jigsaw.w3.org/css-validator/`
  - This tool can help you make sure that your code is correct and will work on CSS3-compliant browsers.

Inline Styles

- Inline style
  - declare an individual element's format using the HTML5 attribute `style`.
- Each CSS property is followed by a colon and the value of the attribute
  - Multiple property declarations are separated by a semicolon
- `Inline styles do not truly separate presentation from content.`
- `To apply similar styles to multiple elements, use embedded style sheets or external style sheets, introduced later in this chapter.`
- `color` property sets text color
  - Color names and hexadecimal codes may be used as the `color` property value.
  - Figure 4.2 contains the HTML standard color set.
  - A list of extended hexadecimal color codes and color names is provided in Appendix B.
  - You can also find a complete list of HTML standard and extended colors at
    - `www.w3.org/TR/css3-color/`



Color name	Value	Color name	Value
aqua	#00FFFF	navy	#000080
black	#000000	olive	#808000
blue	#0000FF	purple	#800080
fuchsia	#FF00FF	red	#FF0000
gray	#808080	silver	#C0C0C0
green	#008000	teal	#008080
lime	#00FF00	yellow	#FFFF00
maroon	#800000	white	#FFFFFF

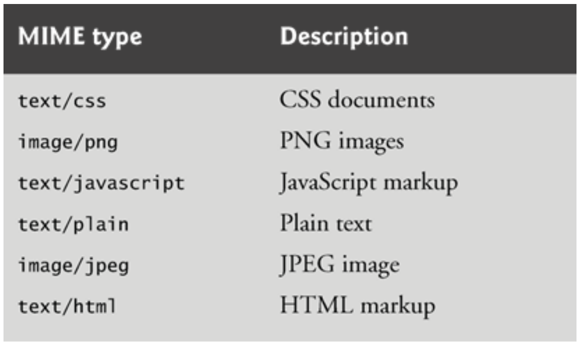
Fig. 4.2 | HTML standard colors and hexadecimal RGB values.

Embedded Style Sheets

- A second technique for using style sheets is **embedded style sheets**, which enable you to embed a CSS3 document in an HTML5 document's head section.

The style Element and MIME Types

- Styles that are placed in a `style` element use selectors to apply style elements throughout the entire document
- `style` element `type` attribute specifies the MIME type (the specific encoding format) of the style sheet. Style sheets use `text/css`.
- Figure 4.4 lists common MIME types used in this book. For a complete list, visit:
  - `www.w3schools.com/media/media_mimeref.asp`



MIME type	Description
text/css	CSS documents
image/png	PNG images
text/javascript	JavaScript markup
text/plain	Plain text
image/jpeg	JPEG image
text/html	HTML markup

Embedded Style Sheets

- The style sheet's body declares the **CSS rules** for the style sheet.
- To achieve the separation between the CSS3 code and the HTML5 that it styles, we'll use a **CSS selector** to specify the elements that will be styled according to a rule.
- An **em element** indicates that its contents should be *emphasized*.
- Each rule body in a style sheet is enclosed in curly braces (`{` and `}`).
- `font-weight` property specifies the "boldness" of text. Possible values are:
  - `bold`
  - `normal` (the default)
  - `bolder` (bolder than bold text)
  - `lighter` (lighter than normal text)
  - Boldness also can be specified with multiples of 100, from 100 to 900 (e.g., 100, 200, ..., 900). Text specified as normal is equivalent to 400, and bold text is equivalent to 700

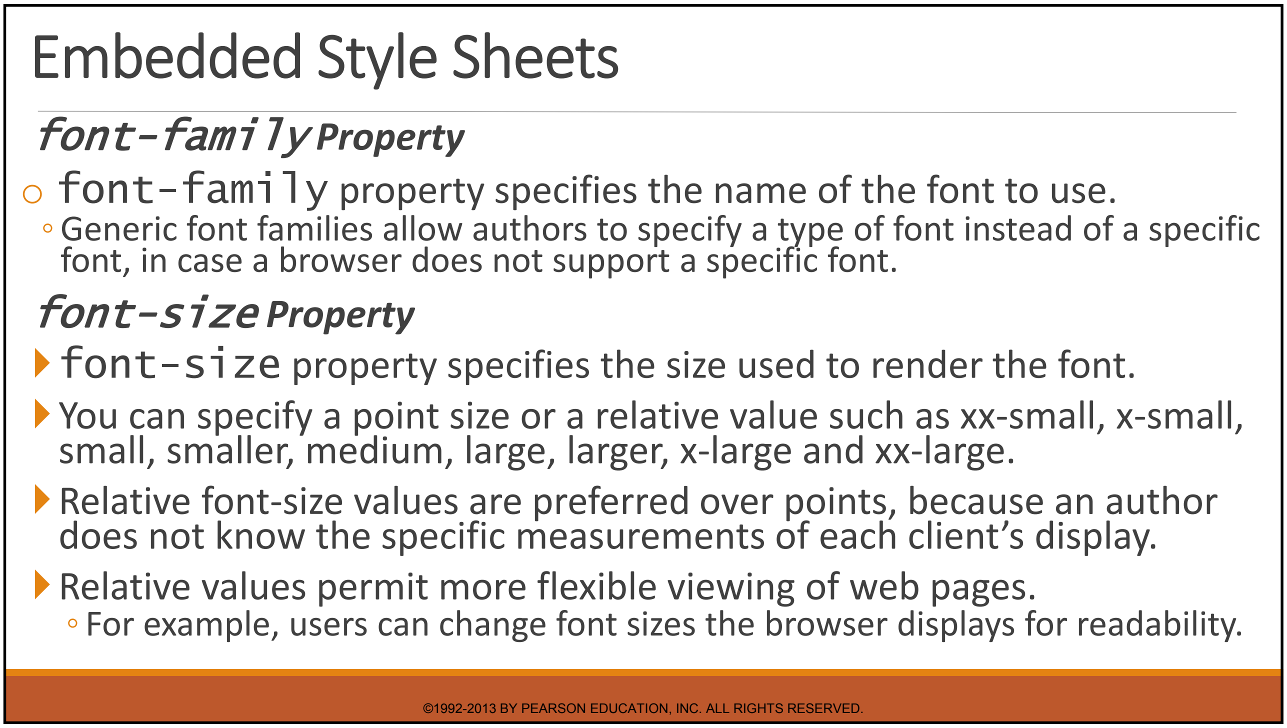
Embedded Style Sheets

Style Classes

- Style-class declarations are preceded by a period (.).
- They define styles that can be applied to *any* element.
- In this example, class special sets color to purple.
- You can also declare `id` selectors.
- If an element in your page has an `id`, you can declare a selector of the form *#elementId* to specify that element's style.

Applying a Style Class

- In many cases, the styles applied to an element (the **parent** or **ancestor element**) also apply to the element's *nested elements* (**child** or **descendant elements**).
- Multiple values of one property can be set or inherited on the same element, so the browser must reduce them to one value for that property per element before they're rendered.
- We discuss the rules for resolving these conflicts in the next section.



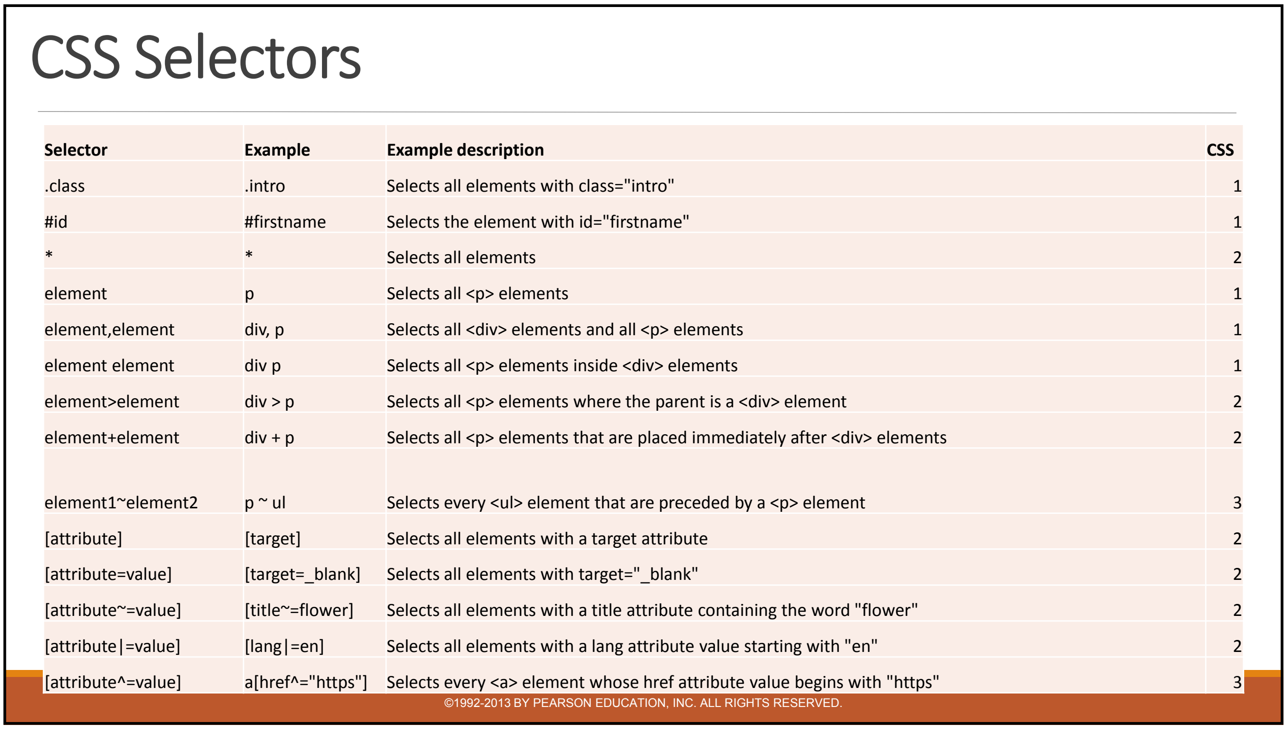
Embedded Style Sheets

font-family Property

- font-family property specifies the name of the font to use.
  - Generic font families allow authors to specify a type of font instead of a specific font, in case a browser does not support a specific font.

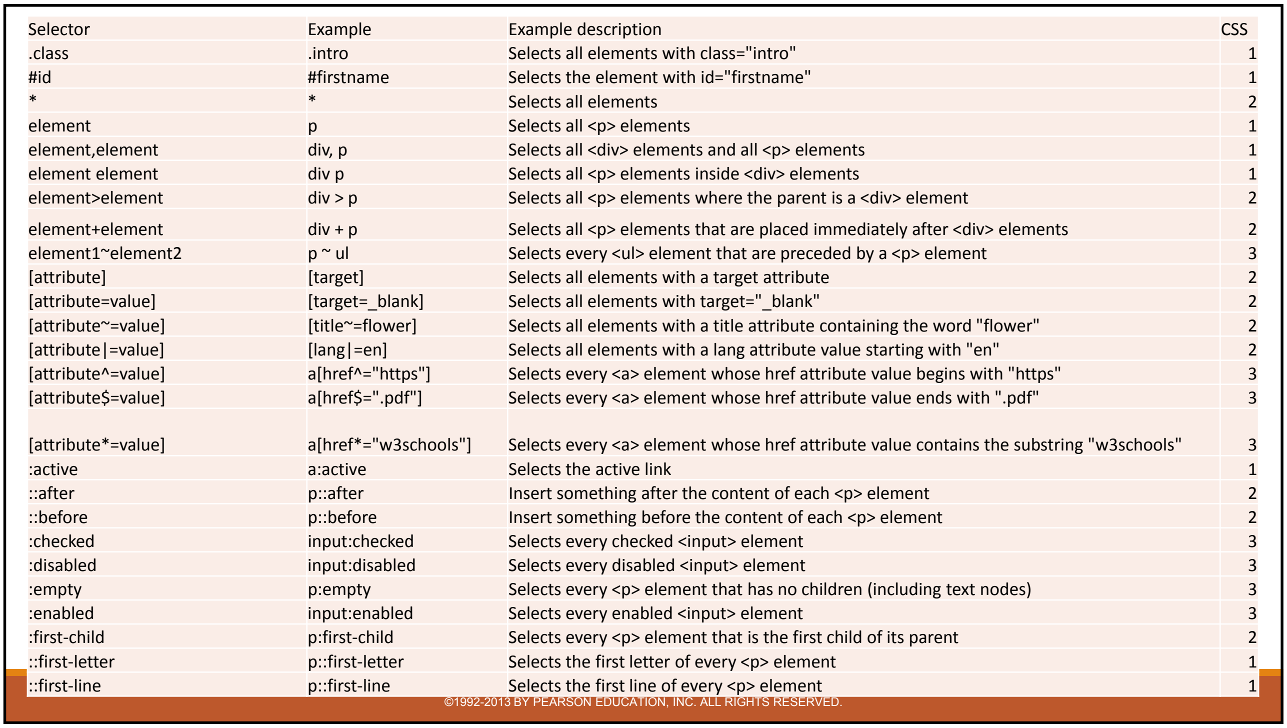
font-size Property

- font-size property specifies the size used to render the font.
- You can specify a point size or a relative value such as xx-small, x-small, small, smaller, medium, large, larger, x-large and xx-large.
- Relative font-size values are preferred over points, because an author does not know the specific measurements of each client’s display.
- Relative values permit more flexible viewing of web pages.
  - For example, users can change font sizes the browser displays for readability.

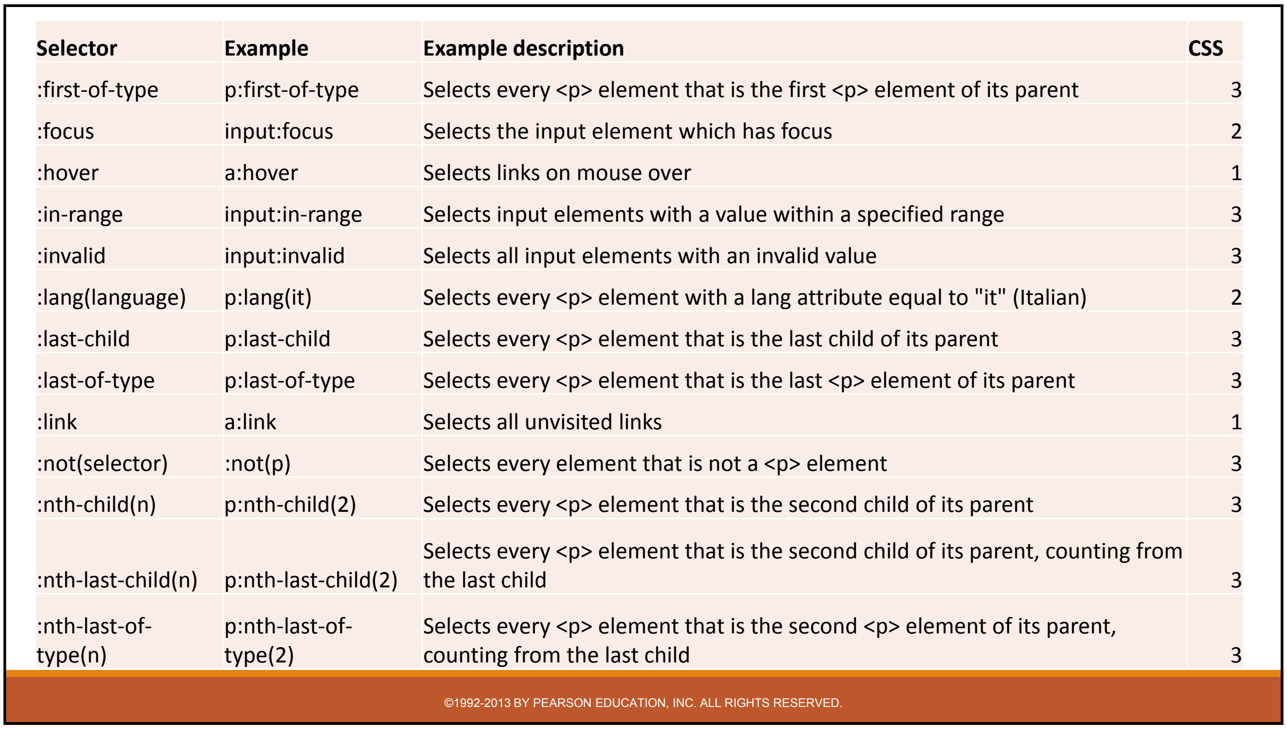


CSS Selectors

Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
element	p	Selects all <p> elements	1
element,element	div, p	Selects all <div> elements and all <p> elements	1
element element	div p	Selects all <p> elements inside <div> elements	1
element>element	div > p	Selects all <p> elements where the parent is a <div> element	2
element+element	div + p	Selects all <p> elements that are placed immediately after <div> elements	2
element1~element2	p ~ ul	Selects every <ul> element that are preceded by a <p> element	3
[attribute]	[target]	Selects all elements with a target attribute	2
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"	2
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"	2
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"	2
[attribute^=value]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"	3



Selector	Example	Example description	CSS
.class	.intro	Selects all elements with class="intro"	1
#id	#firstname	Selects the element with id="firstname"	1
*	*	Selects all elements	2
element	p	Selects all <p> elements	1
element,element	div, p	Selects all <div> elements and all <p> elements	1
element element	div p	Selects all <p> elements inside <div> elements	1
element>element	div > p	Selects all <p> elements where the parent is a <div> element	2
element+element	div + p	Selects all <p> elements that are placed immediately after <div> elements	2
element1~element2	p ~ ul	Selects every <ul> element that are preceded by a <p> element	3
[attribute]	[target]	Selects all elements with a target attribute	2
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"	2
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"	2
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"	2
[attribute^=value]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"	3
[attribute$=value]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"	3
[attribute*=value]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"	3
:active	a:active	Selects the active link	1
::after	p::after	Insert something after the content of each <p> element	2
::before	p::before	Insert something before the content of each <p> element	2
:checked	input:checked	Selects every checked <input> element	3
:disabled	input:disabled	Selects every disabled <input> element	3
:empty	p:empty	Selects every <p> element that has no children (including text nodes)	3
:enabled	input:enabled	Selects every enabled <input> element	3
:first-child	p:first-child	Selects every <p> element that is the first child of its parent	2
::first-letter	p::first-letter	Selects the first letter of every <p> element	1
::first-line	p::first-line	Selects the first line of every <p> element	1



Selector	Example	Example description	CSS
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent	3
:focus	input:focus	Selects the input element which has focus	2
:hover	a:hover	Selects links on mouse over	1
:in-range	input:in-range	Selects input elements with a value within a specified range	3
:invalid	input:invalid	Selects all input elements with an invalid value	3
:lang(language)	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)	2
:last-child	p:last-child	Selects every <p> element that is the last child of its parent	3
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent	3
:link	a:link	Selects all unvisited links	1
:not(selector)	:not(p)	Selects every element that is not a <p> element	3
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent	3
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child	3
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child	3

Selector	Example	Example description	CSS
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent	3
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent	3
:only-child	p:only-child	Selects every <p> element that is the only child of its parent	3
:optional	input:optional	Selects input elements with no "required" attribute	3
:out-of-range	input:out-of-range	Selects input elements with a value outside a specified range	3
:read-only	input:read-only	Selects input elements with the "readonly" attribute specified	3
:read-write	input:read-write	Selects input elements with the "readonly" attribute NOT specified	3
:required	input:required	Selects input elements with the "required" attribute specified	3
:root	:root	Selects the document's root element	3
::selection	::selection	Selects the portion of an element that is selected by a user	
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)	3
:valid	input:valid	Selects all input elements with a valid value	3
:visited	a:visited	Selects all visited links	1

Conflicting Styles

- Styles may be defined by a **user**, an **author** or a **user agent**.
  - Styles **cascade** (and hence the term “Cascading Style Sheets”), or flow together, such that the ultimate appearance of elements on a page results from combining styles defined in several ways.
  - Styles defined by the user take precedence over styles defined by the user agent.
  - Styles defined by authors take precedence over styles defined by the user.
- Most styles defined for parent elements are also **inherited** by child (nested) elements.

○ `text-decoration` property applies decorations to text in an element

`underline`

`overline`

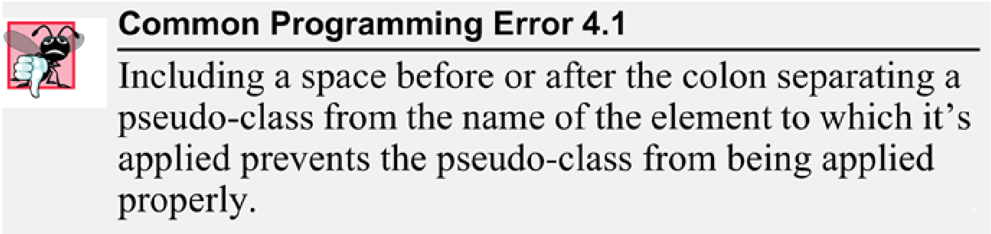
Conflicting Styles

```
line-through
Blink
```

- An example of inheritance in which a child `em` element inherits the `font-size` property from its parent `p` element.
- However, the child `em` element has a `color` property that conflicts with (i.e., has a different value than) the `color` property of its parent p element.
- Properties defined for child and descendant elements have a higher specificity than properties defined for parent and ancestor elements.
- Conflicts are resolved in favor of properties with a higher specificity, **so the child's styles take precedence**.

○ Pseudoclasses give you access to content that's not declared in the document.

○ *hover* pseudoclass is activated when the user moves the mouse cursor over an element.



Common Programming Error 4.1

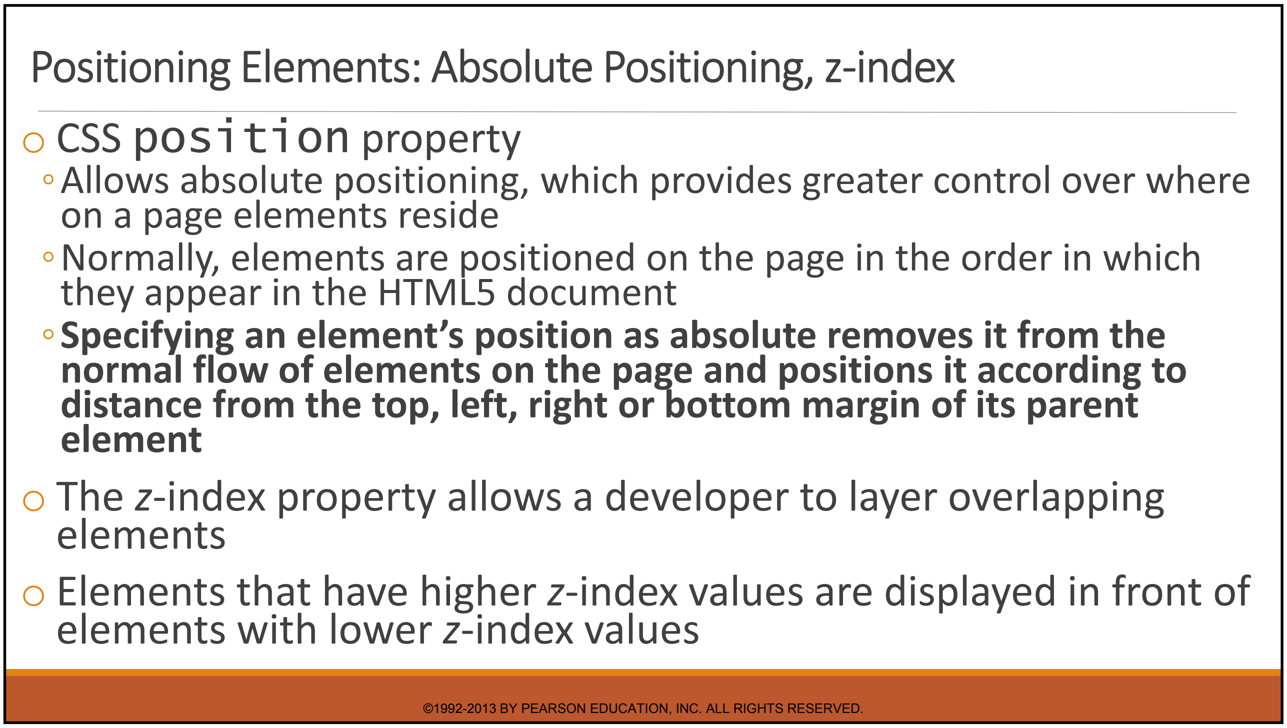
Including a space before or after the colon separating a pseudo-class from the name of the element to which it's applied prevents the pseudo-class from being applied properly.

Conflicting Styles

- Whenever possible, use relative length measurements.
- If you use absolute-length measurements, your document may not scale well on some client browsers (e.g. smartphones)
- Relative length measurements:
  - `px` (pixels – size varies depending on screen resolution)
  - `em` (usually the height of a font’s uppercase M)
  - `ex` (usually the height of a font’s lowercase x)
  - Percentages (of the font’s default size)
- Absolute-length measurements (units that do not vary in size):
  - in (inches)
  - cm (centimeters)
  - mm (millimeters)
  - pt (points; 1 pt = 1/72 in)
  - pc (picas; 1 pc = 12 pt)

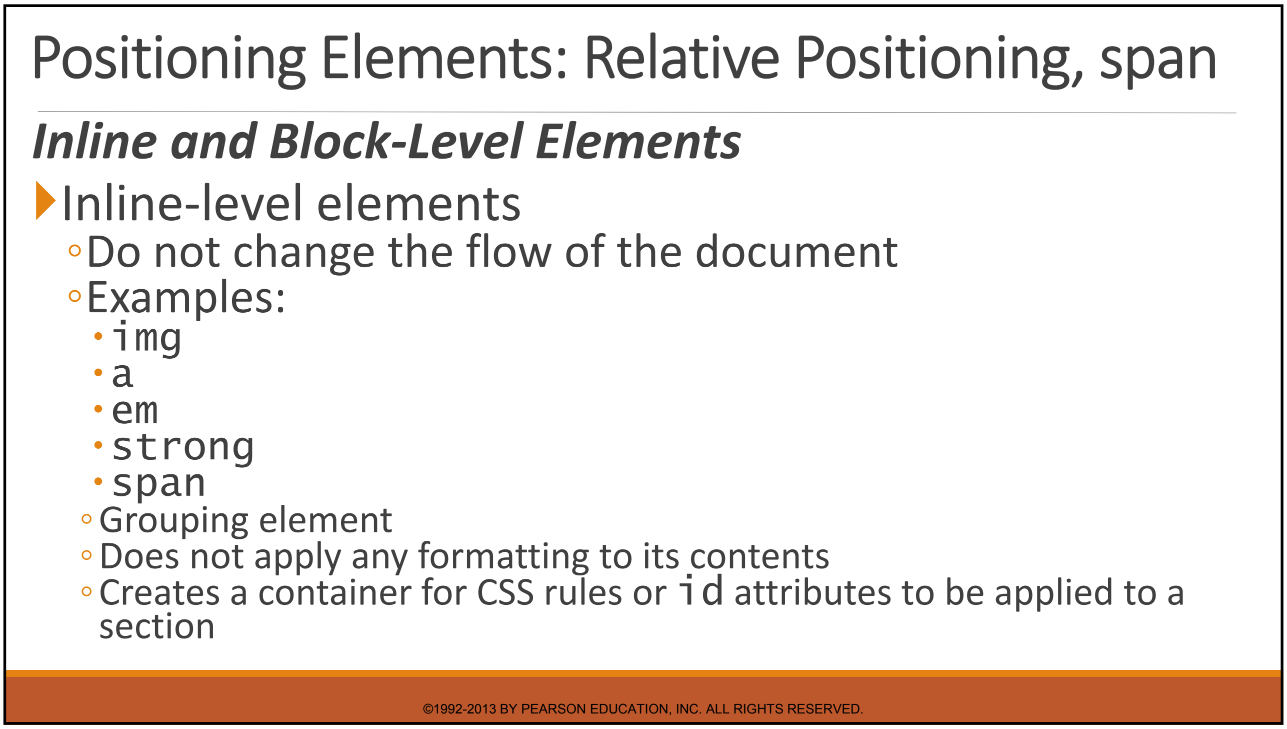
Linking External Style Sheets

- External style sheets are separate documents that contain only CSS rules.
- Help create a uniform look for a website
  - Separate pages can all use the same styles.
  - Modifying a single style-sheet file makes changes to styles across an entire website (or to a portion of one).
- When changes to the styles are required, you need to modify only a single CSS file to make style changes across *all* the pages that use those styles. This concept is sometimes known as **skinning**.
- **CSS comments** may be placed in any type of CSS code (i.e., inline styles, embedded style sheets and external style sheets) and always start with /* and end with */. .
- **`link` element**
  - Uses `rel` attribute to specify a relationship between two documents
  - `rel` attribute declares the linked document to be a stylesheet for the document
- **`type` attribute specifies the MIME type of the related document**
- `href` attribute provides the URL for the document containing the style sheet
- `<link rel="stylesheet" type="text/css" href="styles.css">`



Positioning Elements: Absolute Positioning, z-index

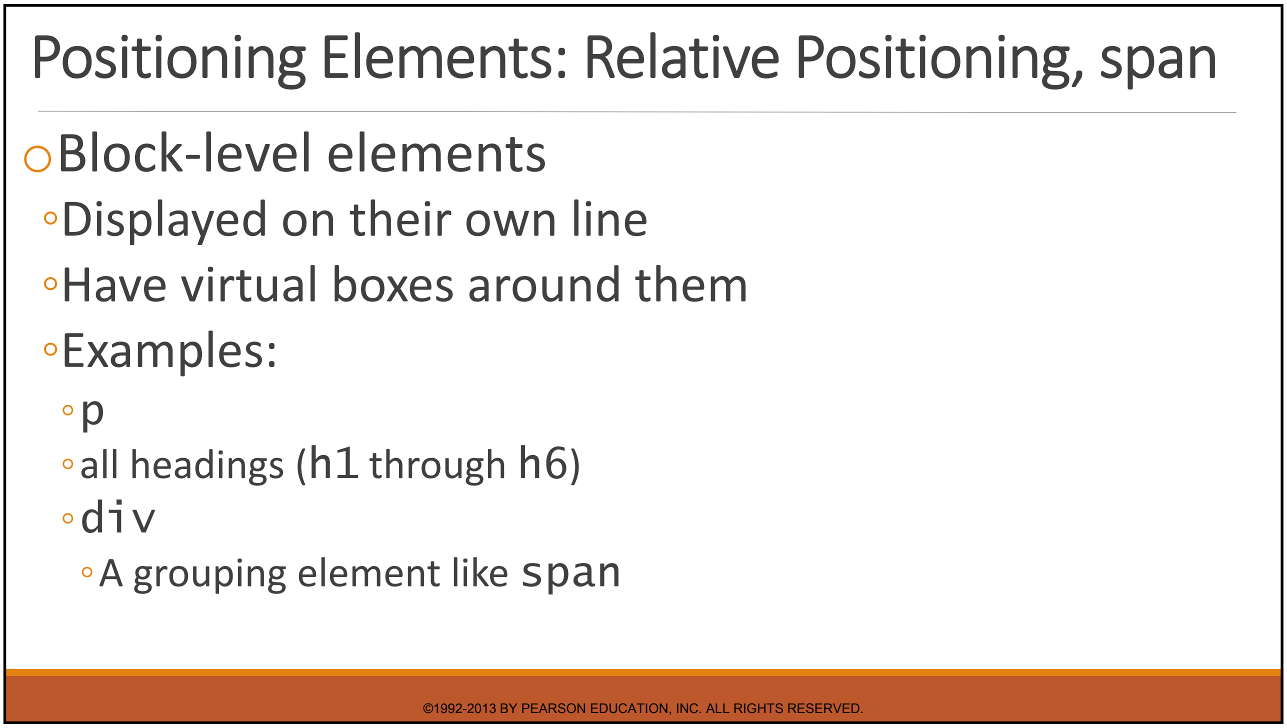
- CSS `position` property
  - Allows absolute positioning, which provides greater control over where on a page elements reside
  - Normally, elements are positioned on the page in the order in which they appear in the HTML5 document
  - **Specifying an element’s position as absolute removes it from the normal flow of elements on the page and positions it according to distance from the top, left, right or bottom margin of its parent element**
- The *z*-index property allows a developer to layer overlapping elements
- Elements that have higher *z*-index values are displayed in front of elements with lower *z*-index values



Positioning Elements: Relative Positioning, span

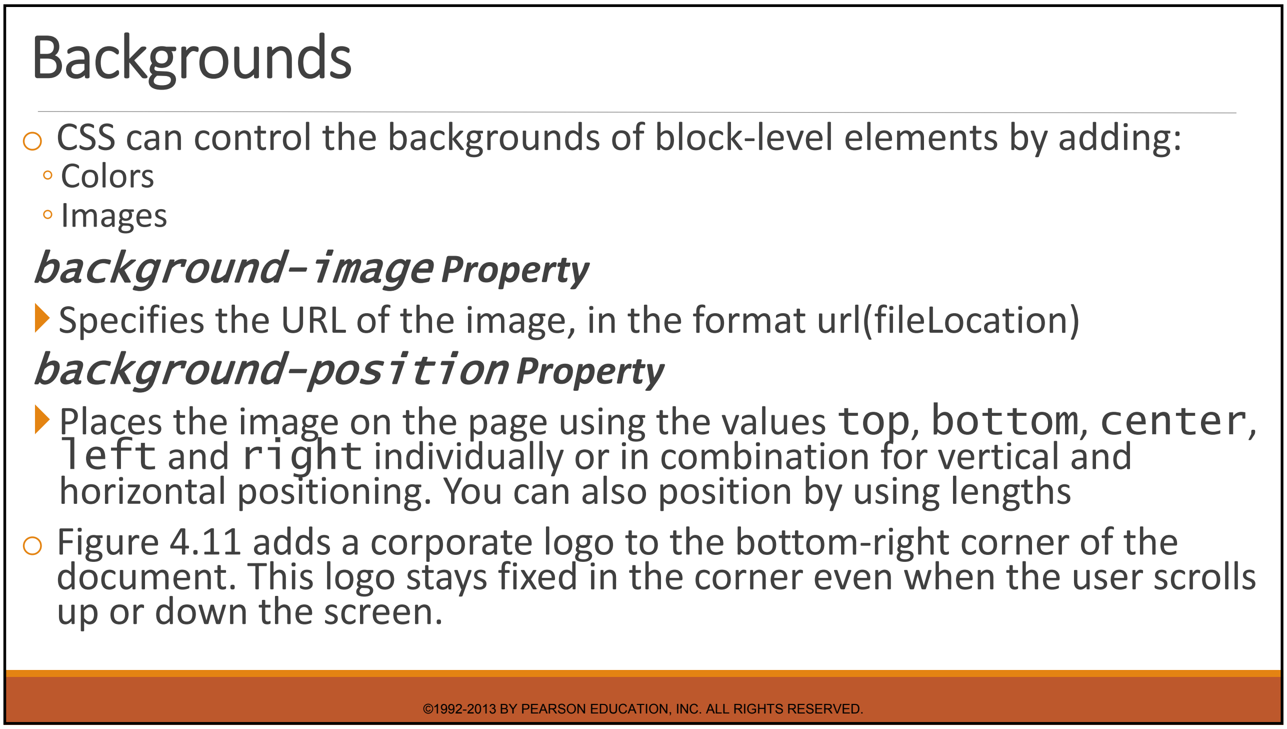
Inline and Block-Level Elements

- Inline-level elements
  - Do not change the flow of the document
  - Examples:
    - `img`
    - `a`
    - `em`
    - `strong`
    - `span`
      - Grouping element
      - Does not apply any formatting to its contents
      - Creates a container for CSS rules or `id` attributes to be applied to a section



Positioning Elements: Relative Positioning, span

- Block-level elements
  - Displayed on their own line
  - Have virtual boxes around them
  - Examples:
    - `p`
    - all headings (`h1` through `h6`)
    - `div`
      - A grouping element like `span`



Backgrounds

- CSS can control the backgrounds of block-level elements by adding:
  - Colors
  - Images

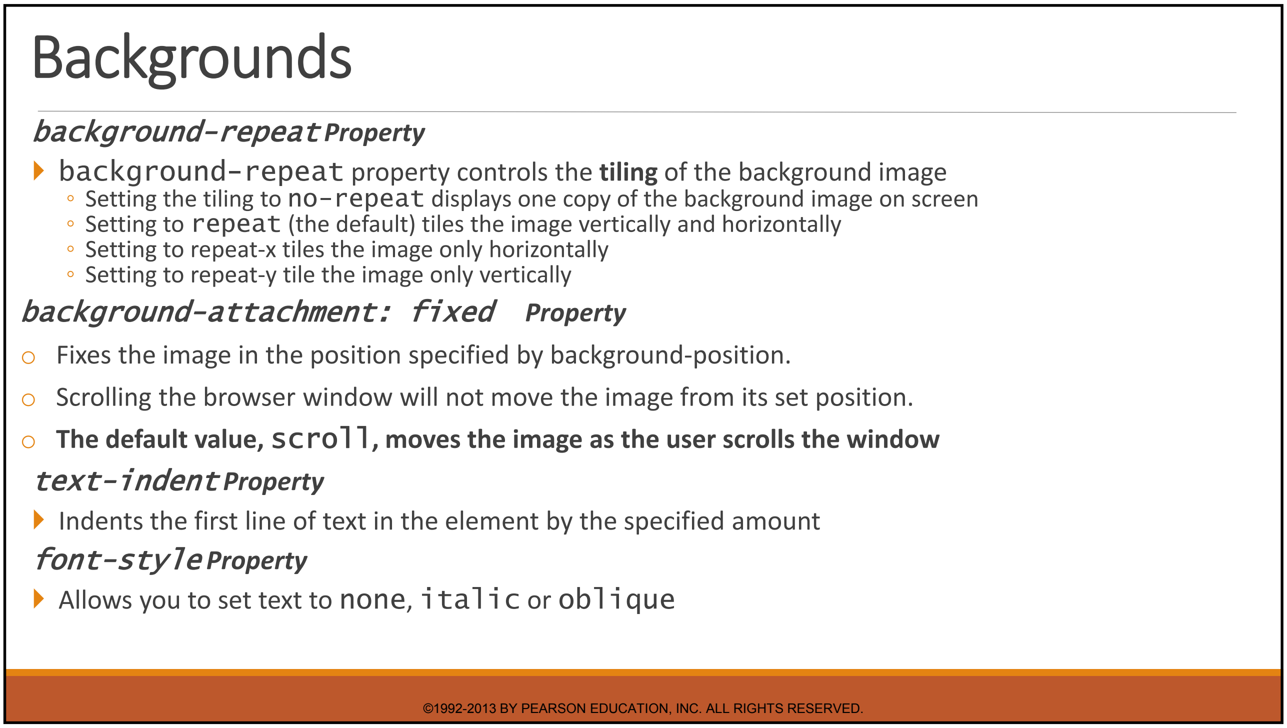
***`background-image`* Property**

- Specifies the URL of the image, in the format url(fileLocation)

***`background-position`* Property**

- Places the image on the page using the values `top`, `bottom`, `center`, `left` and `right` individually or in combination for vertical and horizontal positioning. You can also position by using lengths

- Figure 4.11 adds a corporate logo to the bottom-right corner of the document. This logo stays fixed in the corner even when the user scrolls up or down the screen.



Backgrounds

***`background-repeat`* Property**

- `background-repeat` property controls the **tiling** of the background image
  - Setting the tiling to `no-repeat` displays one copy of the background image on screen
  - Setting to `repeat` (the default) tiles the image vertically and horizontally
  - Setting to repeat-x tiles the image only horizontally
  - Setting to repeat-y tile the image only vertically

***`background-attachment: fixed`* Property**

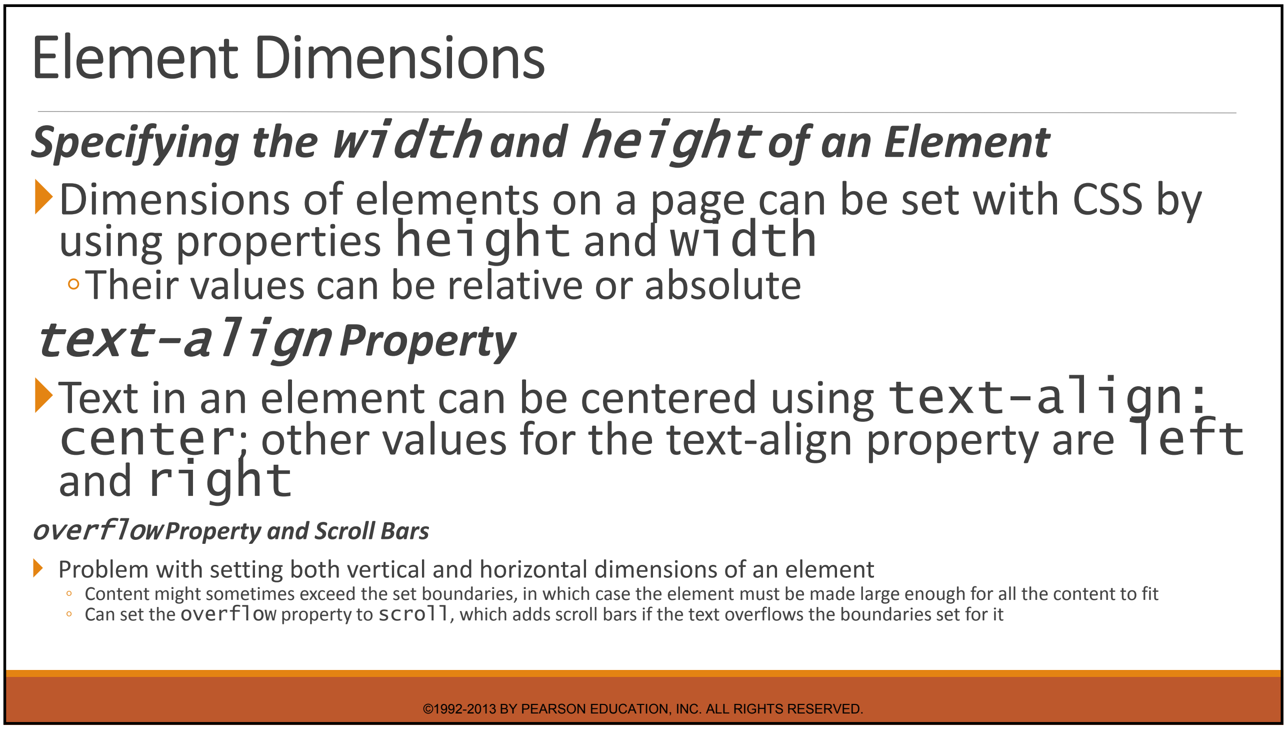
- Fixes the image in the position specified by background-position.
- Scrolling the browser window will not move the image from its set position.
- **The default value, `scroll`, moves the image as the user scrolls the window**

***`text-indent`* Property**

- Indents the first line of text in the element by the specified amount

***`font-style`* Property**

- Allows you to set text to `none`, `italic` or `oblique`



Element Dimensions

Specifying the `width` and `height` of an Element

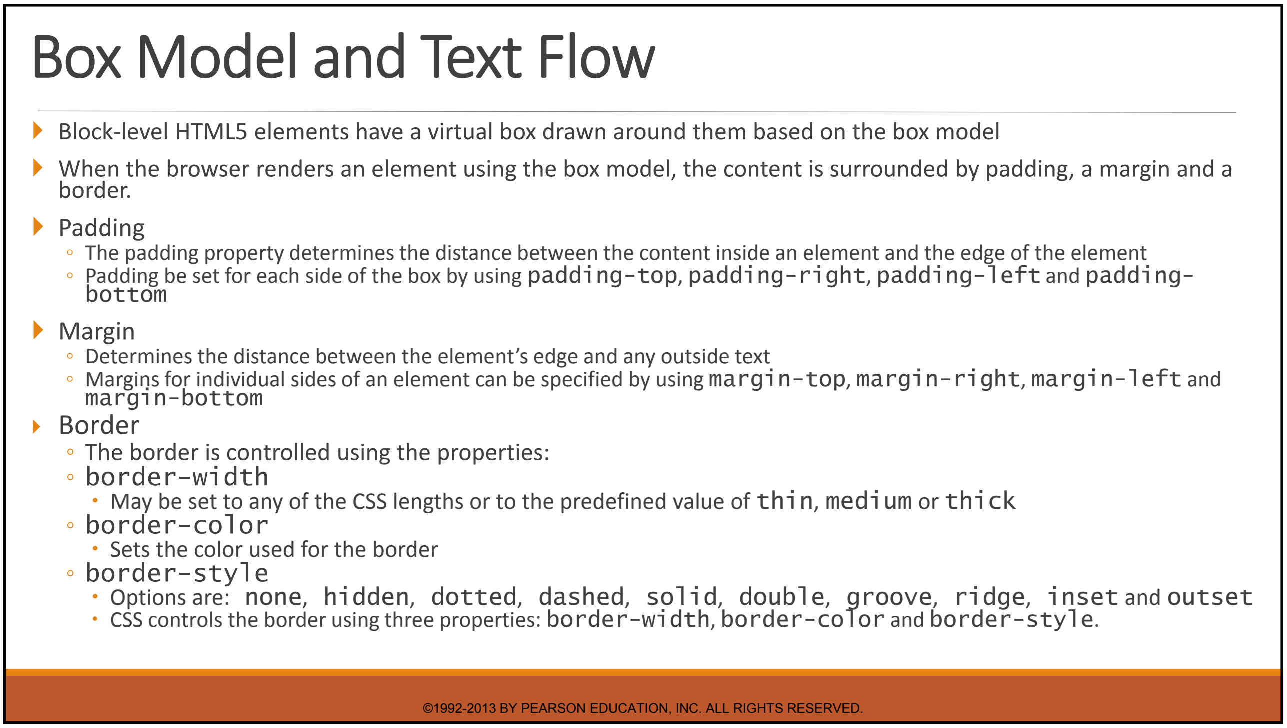
- Dimensions of elements on a page can be set with CSS by using properties `height` and `width`
  - Their values can be relative or absolute

***`text-align`* Property**

- Text in an element can be centered using `text-align: center`; other values for the text-align property are `left` and `right`

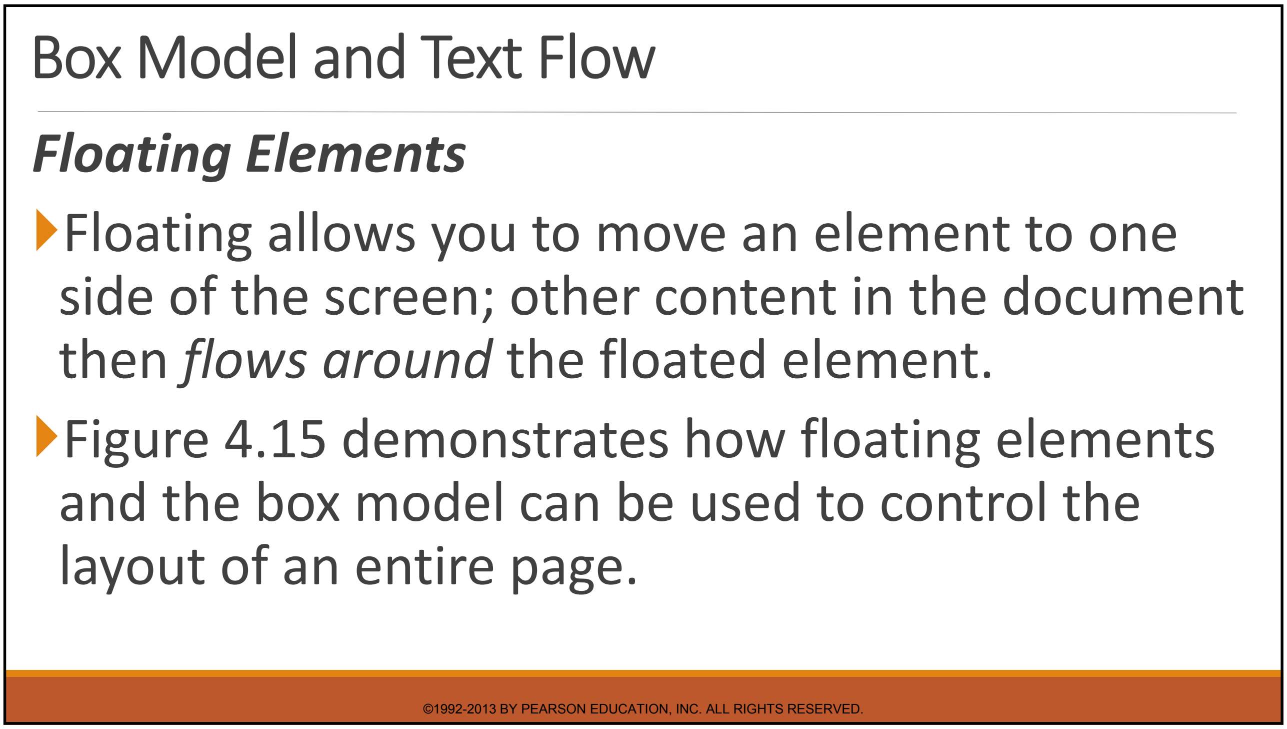
***`overflow`* Property and Scroll Bars**

- Problem with setting both vertical and horizontal dimensions of an element
  - Content might sometimes exceed the set boundaries, in which case the element must be made large enough for all the content to fit
  - Can set the `overflow` property to `scroll`, which adds scroll bars if the text overflows the boundaries set for it



Box Model and Text Flow

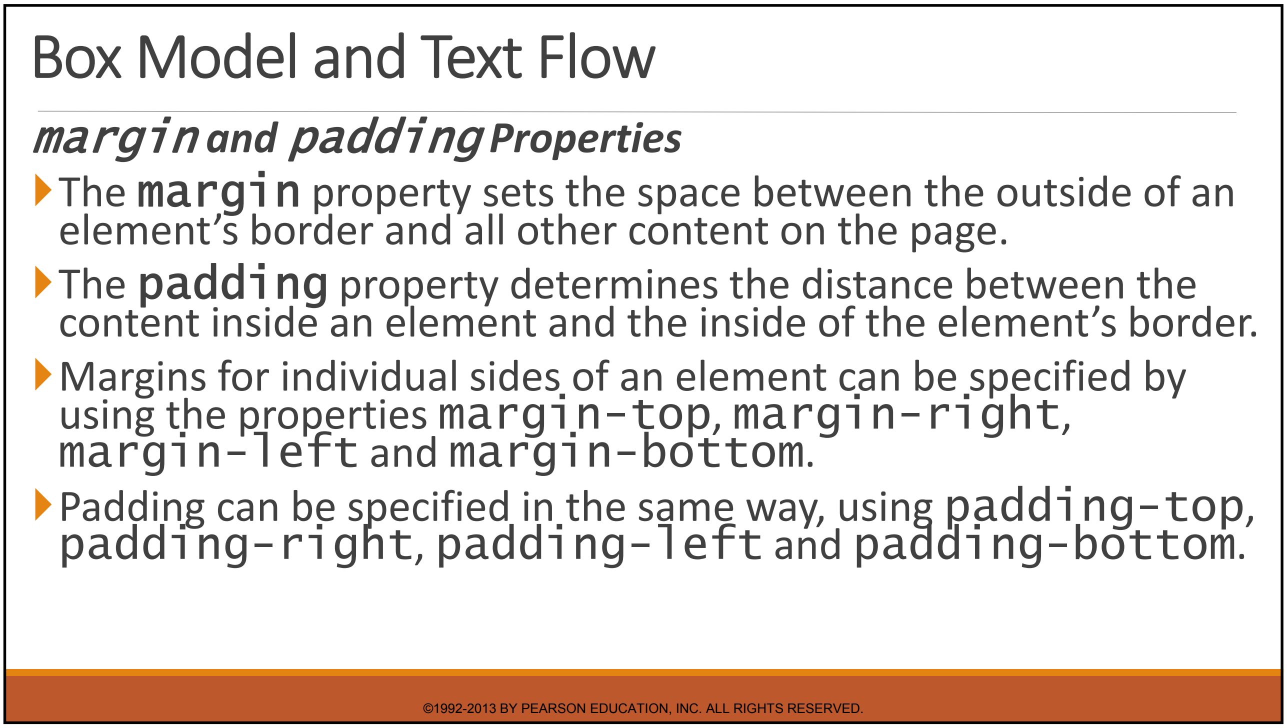
- Block-level HTML5 elements have a virtual box drawn around them based on the box model
- When the browser renders an element using the box model, the content is surrounded by padding, a margin and a border.
- Padding
  - The padding property determines the distance between the content inside an element and the edge of the element
  - Padding be set for each side of the box by using `padding-top`, `padding-right`, `padding-left` and `padding-bottom`
- Margin
  - Determines the distance between the element's edge and any outside text
  - Margins for individual sides of an element can be specified by using `margin-top`, `margin-right`, `margin-left` and `margin-bottom`
- Border
  - The border is controlled using the properties:
  - `border-width`
    - May be set to any of the CSS lengths or to the predefined value of `thin`, `medium` or `thick`
  - `border-color`
    - Sets the color used for the border
  - `border-style`
    - Options are: `none`, `hidden`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset` and `outset`
    - CSS controls the border using three properties: `border-width`, `border-color` and `border-style`.



Box Model and Text Flow

Floating Elements

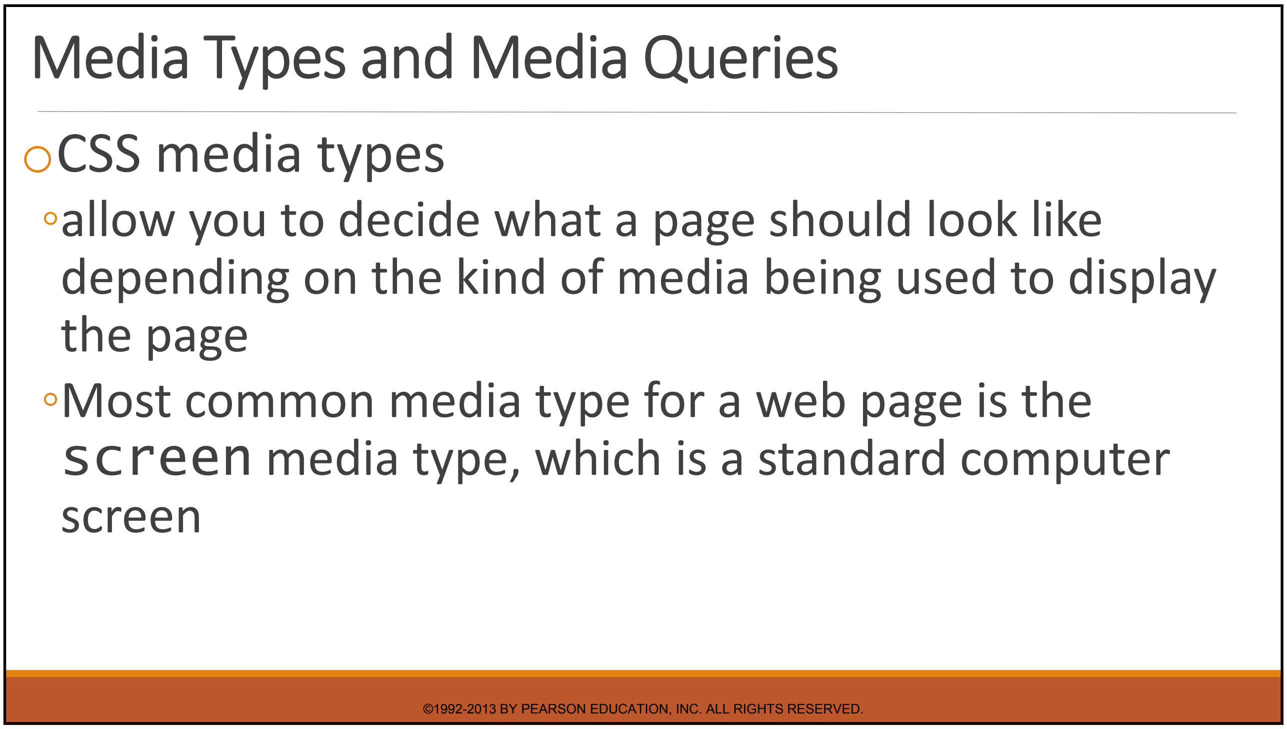
- Floating allows you to move an element to one side of the screen; other content in the document then *flows around* the floated element.
- Figure 4.15 demonstrates how floating elements and the box model can be used to control the layout of an entire page.



Box Model and Text Flow

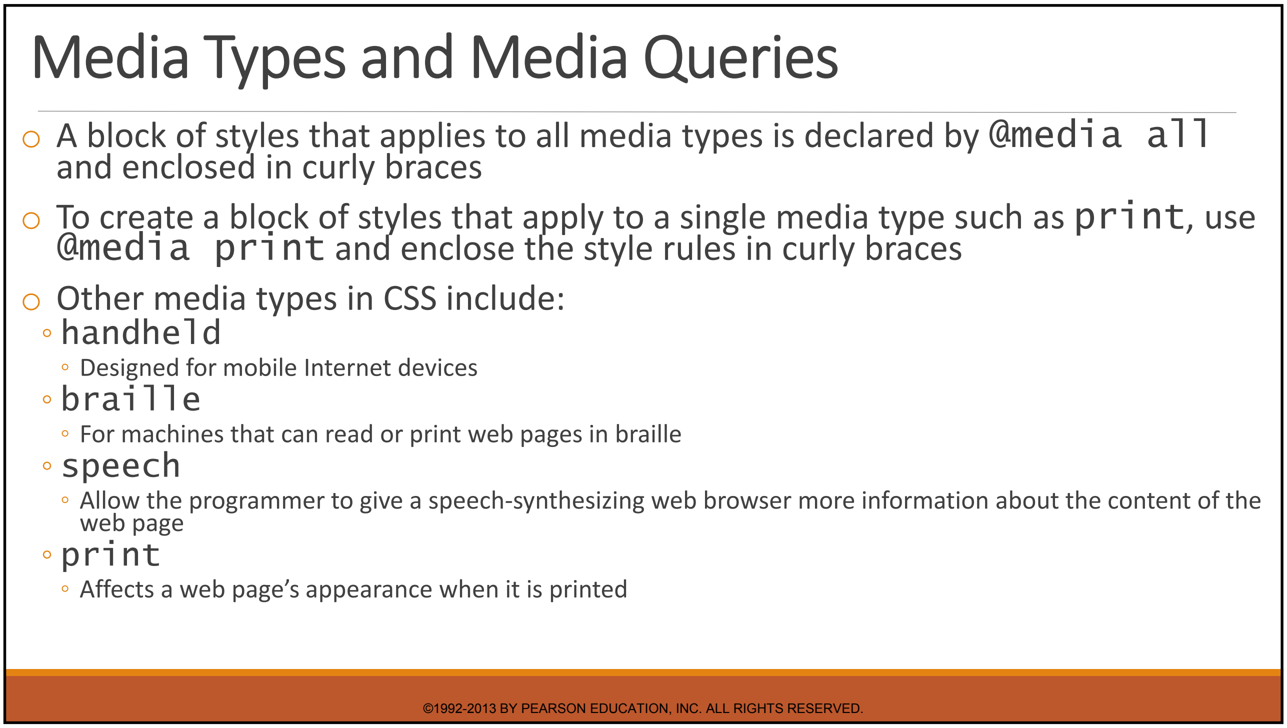
`margin` and *`padding`* Properties

- The `margin` property sets the space between the outside of an element’s border and all other content on the page.
- The `padding` property determines the distance between the content inside an element and the inside of the element’s border.
- Margins for individual sides of an element can be specified by using the properties `margin-top`, `margin-right`, `margin-left` and `margin-bottom`.
- Padding can be specified in the same way, using `padding-top`, `padding-right`, `padding-left` and `padding-bottom`.



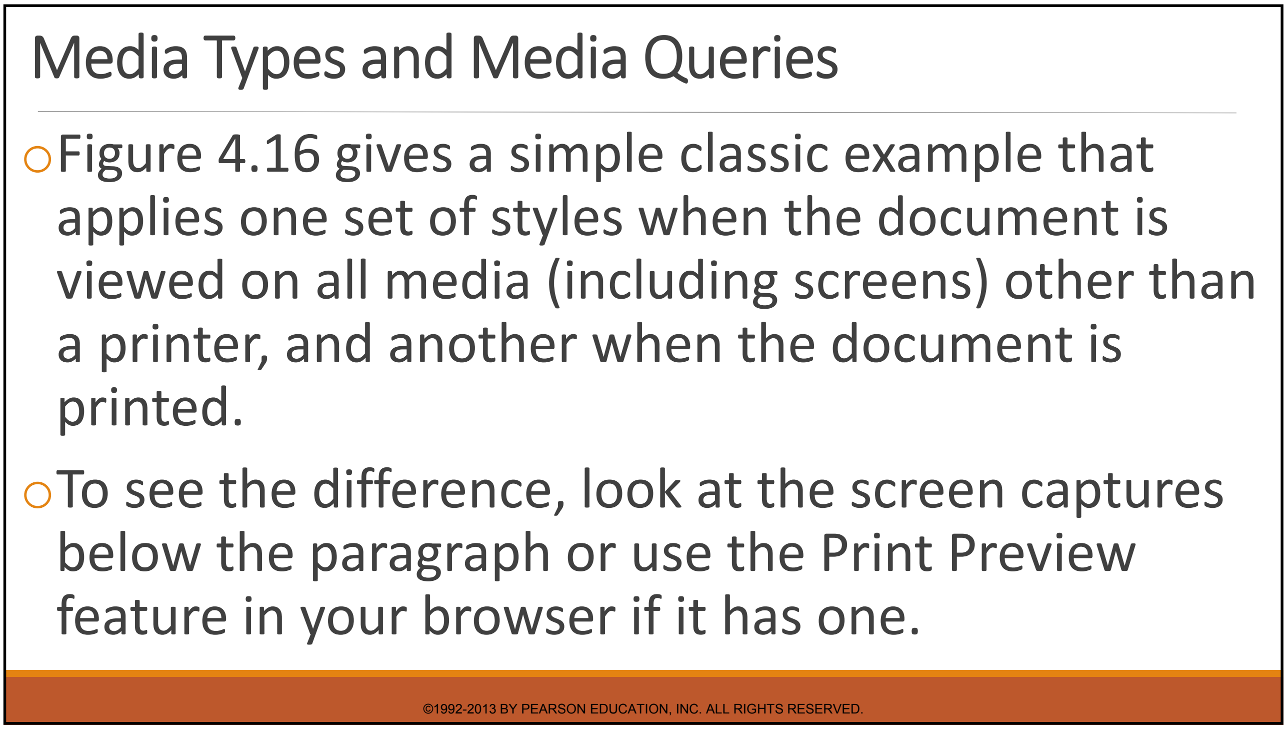
Media Types and Media Queries

- CSS media types
  - allow you to decide what a page should look like depending on the kind of media being used to display the page
  - Most common media type for a web page is the `screen` media type, which is a standard computer screen



Media Types and Media Queries

- A block of styles that applies to all media types is declared by `@media all` and enclosed in curly braces
- To create a block of styles that apply to a single media type such as `print`, use `@media print` and enclose the style rules in curly braces
- Other media types in CSS include:
  - `handheld`
    - Designed for mobile Internet devices
  - `braille`
    - For machines that can read or print web pages in braille
  - `speech`
    - Allow the programmer to give a speech-synthesizing web browser more information about the content of the web page
  - `print`
    - Affects a web page's appearance when it is printed



Media Types and Media Queries

- Figure 4.16 gives a simple classic example that applies one set of styles when the document is viewed on all media (including screens) other than a printer, and another when the document is printed.
- To see the difference, look at the screen captures below the paragraph or use the Print Preview feature in your browser if it has one.

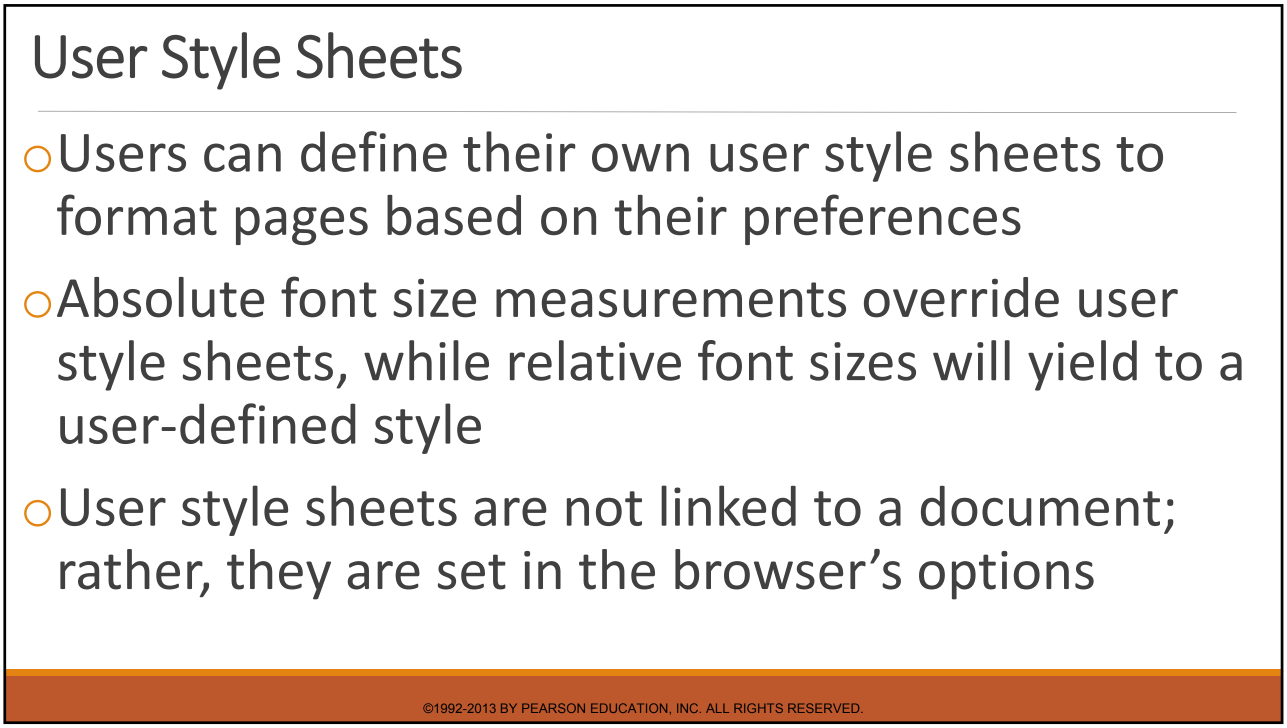
Media Types and Media Queries

Media Queries

- Allow you to format your content to specific output devices.
- Include a media type and expressions that check the media features of the output device.
- Common media features include:
  - `width`—the width of the part of the screen on which the document is rendered, including any scrollbars
  - `height`—the height of the part of the screen on which the document is rendered, including any scrollbars
  - `device-width`—the width of the screen of the output device
  - device-height—the height of the screen of the output device
  - `orientation`—if the height is greater than the width, orientation is portrait, and if the width is greater than the height, orientation is landscape
  - `aspect-ratio`—the ratio of width to height
  - `device-aspect-ratio`—the ratio of device-width to device-height

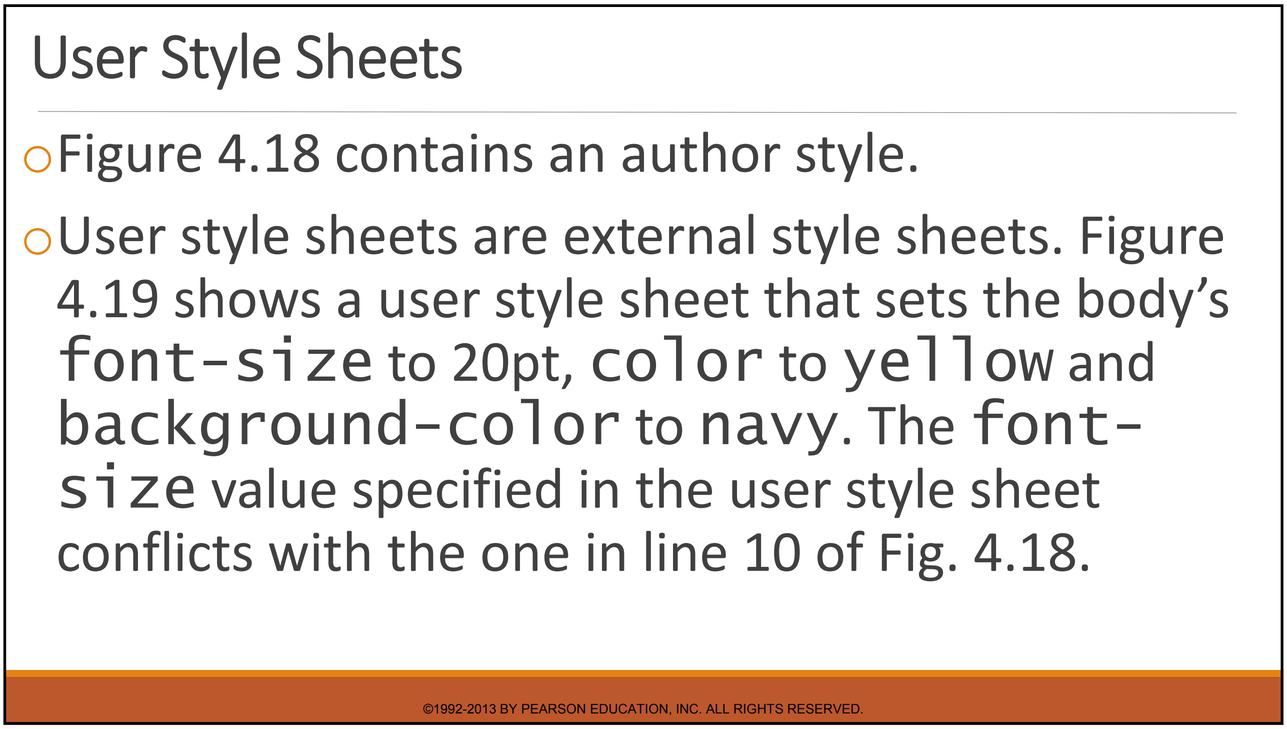
Drop-Down Menus

- :hover pseudoclass
  - used to apply styles to an element when the mouse cursor is over it
- display property
  - allows a programmer to decide if an element is displayed as a block element, inline element, or is not rendered at all (none)



User Style Sheets

- Users can define their own user style sheets to format pages based on their preferences
- Absolute font size measurements override user style sheets, while relative font sizes will yield to a user-defined style
- User style sheets are not linked to a document; rather, they are set in the browser's options



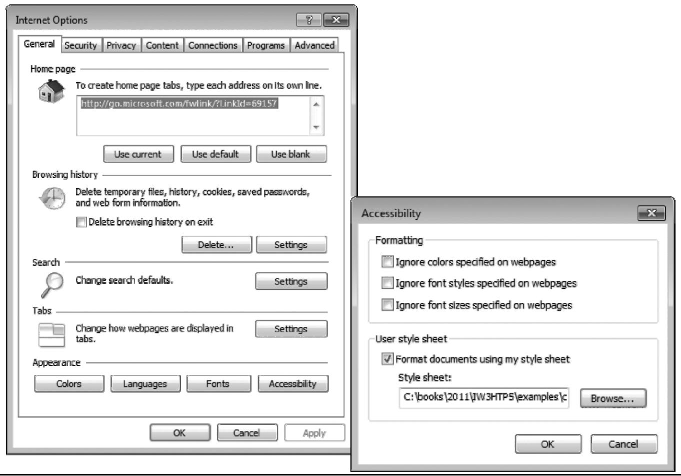
User Style Sheets

- Figure 4.18 contains an author style.
- User style sheets are external style sheets. Figure 4.19 shows a user style sheet that sets the body's `font-size` to 20pt, `color` to `yellow` and `background-color` to `navy`. The `font-size` value specified in the user style sheet conflicts with the one in line 10 of Fig. 4.18.

User Style Sheets

Adding a User Style Sheet

- User style sheets are not linked to a document; rather, they're set in the browser's options.
- To add a user style sheet in IE9, select **Internet Options...**, located in the **Tools** menu. In the **Internet Options** dialog (Fig. 4.20) that appears, click **Accessibility...**, check the **Format documents using my style sheet** checkbox, and type the location of the user style sheet.
- To add a user style sheet in Firefox, find your Firefox profile using the instructions at
  - www.mozilla.org/support/firefox/profile#locate
- Place a style sheet called `userContent.css` in the `chrome` subdirectory.
- For information on adding a user style sheet in Chrome, see
  - www.google.com/support/forum/p/Chrome/thread?tid=1fa0dd079dbdc2ff&hl=en.



Internet Options

Home page

To create home page tabs, type each address on its own line.

http://go.microsoft.com/fwlink/?LinkId=69157

Browsing history

Delete temporary files, history, cookies, saved passwords, and web form information.

□ Delete browsing history on exit

Search

Change search defaults.

Tabs

Change how webpages are displayed in tabs.

Appearance

Accessibility

Formatting

□ Ignore colors specified on webpages

□ Ignore font styles specified on webpages

□ Ignore font sizes specified on webpages

User style sheet

☑ Format documents using my style sheet

Style sheet: C:\books\2011\IW3HTP5\examples\c

Fig. 4.20 | User style sheet in Internet Explorer 9.