

# Web Design with HTML5 & CSS3 8<sup>th</sup> Edition

## Applying CSS Styles to Webpages



# Objectives

- Explain the importance of separating design from content
- Describe Cascading Style Sheets (CSS)
- Define inline, embedded, and external styles and their order of precedence
- Describe a CSS rule and its syntax
- Explain the difference between a selector, property, and value

# Objectives (continued)

- Create styles that use text and color properties
- Explain the difference between inline and block content
- Describe the CSS box model and how to apply margins, padding, and borders
- Create an external style sheet and link it to an HTML page

# Objectives (continued 1)

- Create styles that use padding, border, and margin properties
- Float an image
- Create styles that use list properties
- Add comments to an external style sheet
- Validate a CSS file

# Using Cascading Style Sheets

- **Style** – It is a rule that defines the appearance of an element on a webpage
- **Style sheet** – It is the set of CSS style rules
- Style sheets provide a means to separate style from content because it gives the flexibility to redesign or rebrand a website
- A single CSS style sheet file containing the defined styles can be attached to several webpages to apply the styles to all the attached pages

# Inline Styles

- **Inline style** – It is used to add a style to the start tag for an element, such as a heading or paragraph, using the **style attribute**

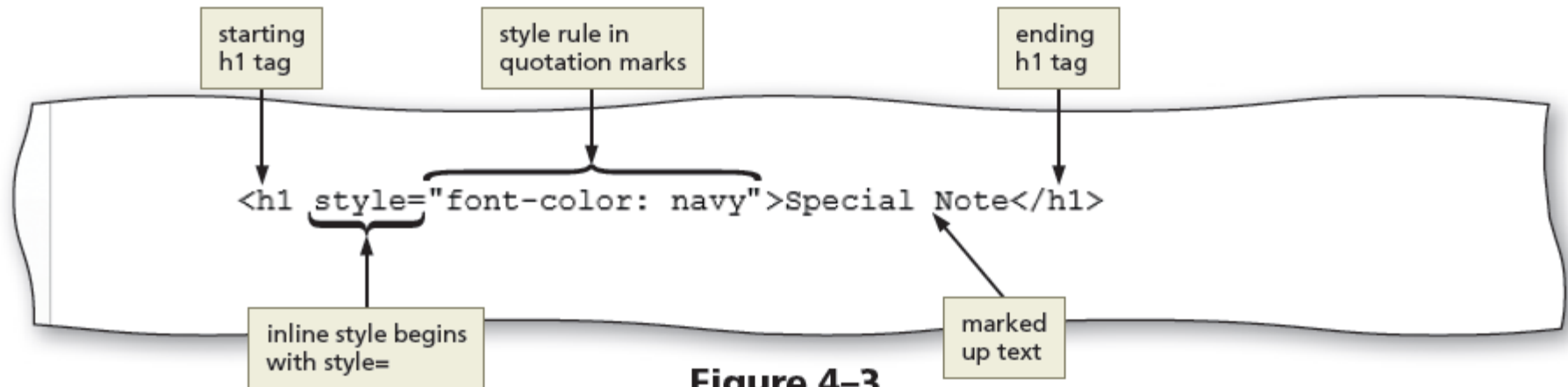


Figure 4-3

# Embedded Style Sheets

- An **embedded style sheet**, also called an **internal style sheet**, includes the style sheet within the opening `<head>` and closing `</head>` tags of the HTML document



Figure 4-4

# External Style Sheets

- An **external style sheet** – It is a CSS file that contains all of the styles that can be applied to more than one page in a website
- **External style sheets** are also called **linked style sheets**
- An external style sheet is a text file with the **.css** extension
- To apply an external style sheet, link it (or attach it) to a webpage using a link in the head section of the webpage



# External Style Sheets (continued)

- External style sheet provides flexibility to quickly change webpage formats because the styles used in it are applied to every page linked to it
- Changing the look of an entire website is sometimes called **reskinning** the website



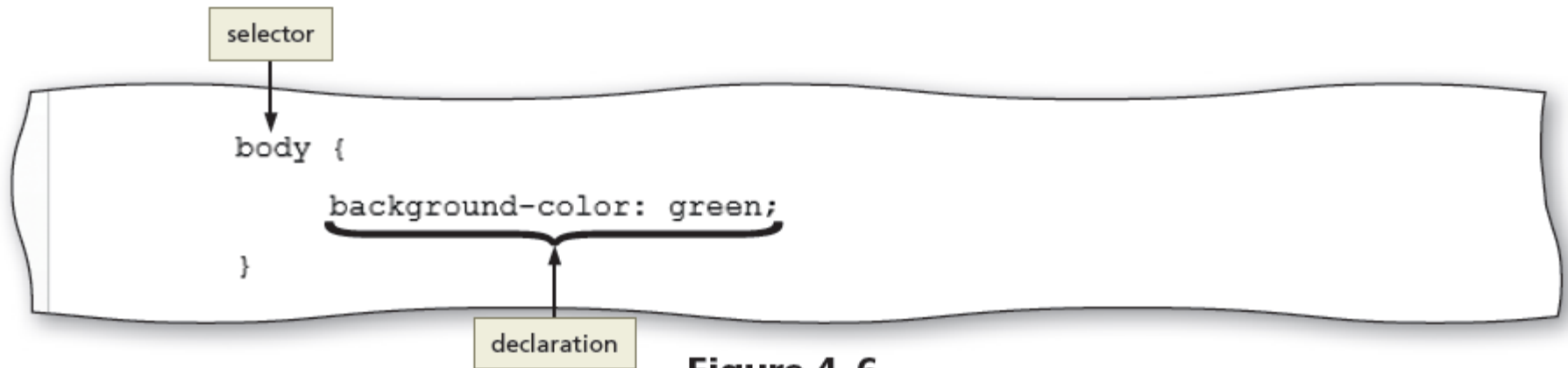
Figure 4-5

# Style Sheet Precedence

- Style sheets are said to “cascade” because each type of style has a specified level of precedence (or priority) in relationship to the others
- CSS properties can be inherited from a parent element through a principle called **inheritance**
- If a selector has more than one CSS rule, **specificity** determines which CSS rule to apply

# CSS Basics

- Each CSS rule consists of a selector and a declaration



**Figure 4-6**

# CSS Basics (continued)

- **Selector** – It is the part of the statement that identifies what to style
- Any HTML5 element such as body, header, nav, main, or footer may be a selector
- A selector may also be the value of an id or class attribute
- The **declaration** defines the exact formatting of the style

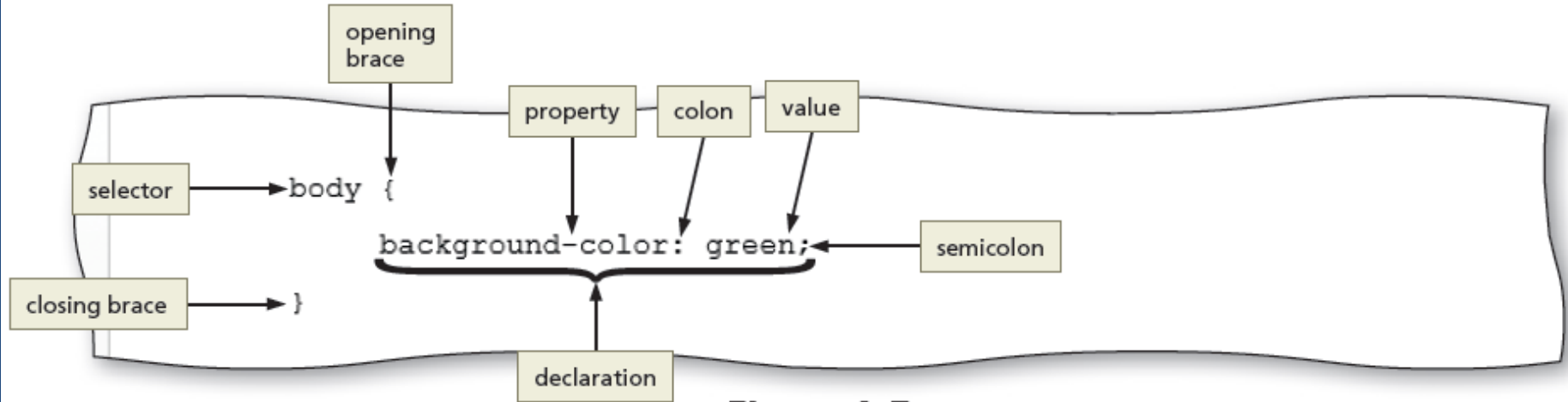
# CSS Basics (continued 1)

- A declaration consists of a property and a value, separated by a colon and followed by a semicolon
- The **property** identifies the style quality or characteristic to apply, such as
  - color (text color)
  - background-color
  - text-indent
  - border-width
  - font-style

# CSS Basics (continued 2)

- For each property, the declaration includes a related **value** that identifies the particular property value to apply

# CSS Basics (continued 3)



**Figure 4-7**

# CSS Text Properties

Table 4–1 Common Text Properties and Values

Property	Description	Common Values
font-family	Specific and general font names	font-family: Cambria, "Times New Roman", serif; font-family: Verdana, Arial, sans-serif; font-family: Georgia, "Times New Roman", serif;
font-size	Absolute or relative size of a font	font-size: 1.5em; font-size: 50%; font-size: x-large; font-size: 14pt;
font-weight	Weight of a font	font-weight: bold; font-weight: bolder; font-weight: lighter;
font-style	Style of a font	font-style: normal; font-style: italic; font-style: oblique;
text-align	Alignment of text	text-align: center; text-align: right; text-align: justify;
color	Color of text	color: red; color: blue; color: green;



# CSS Text Properties (continued)

- **Fallback values** – They are the additional values provided for the font-family property in case the browser does not support the primary font
- CSS measures font sizes using many measurement units, including pixels, points, and ems, and by keyword or percentage

# CSS Text Properties (continued)

## 1)

Table 4–2 Font Size Measurement Units

Unit	Definition	Example	Comments
<b>em</b>	Relative to the default font size of the element	font-size: 1.25em;	Recommended by W3C; sizes are relative to the browser's default font size
%	Relative to the default font size of the element	font-size: 50%;	Recommended by W3C; sizes are relative to the browser's default font size
px	Number of pixels	font-size: 25px;	Depends on screen resolution
pt	Number of points	font-size: 12pt;	Use for printing webpages
keyword	Relative to a limited range of sizes	font-size: xx-small;	Sizes are relative to the browser's default font size, but size options are limited

# CSS Colors

- HTML uses color names or codes to designate color values
- Two types of color codes can be used with CSS:
  - Hexadecimal
  - RGB
- **Hexadecimal** values consist of a six-digit number code that corresponds to **RGB (Red, Green, Blue)** color values

# CSS Colors (continued)

- To use a color in a style rule declaration, use the color value as the property value
- For example, to style a background color as gray use, `background-color: #808080;`

Table 4–3 Color Values

Color	Hexadecimal	RGB
Black	#000000	rgb(0,0,0)
White	#FFFFFF	rgb(255,255,255)
Red	#FF0000	rgb(255,0,0)
Green	#008000	rgb(0,128,0)
Blue	#0000FF	rgb(0,0,255)
Yellow	#FFFF00	rgb(255,255,0)
Orange	#FFA500	rgb(255,165,0)
Gray	#808080	rgb(128,128,128)

# Understanding Inline Elements and Block Elements

- HTML elements are positioned on the webpage as a block or as inline content
- A **block element** appears as a block because it starts and ends with a new line, such as the main element or a paragraph element
- **Inline elements** are displayed without line breaks so they flow within the same line
- Inline content always appears within block elements

# CSS Box Model

- Each block element such as a header, nav, main, and footer element is displayed in a browser as a box with content
- The **CSS box model** describes content boxes on a webpage

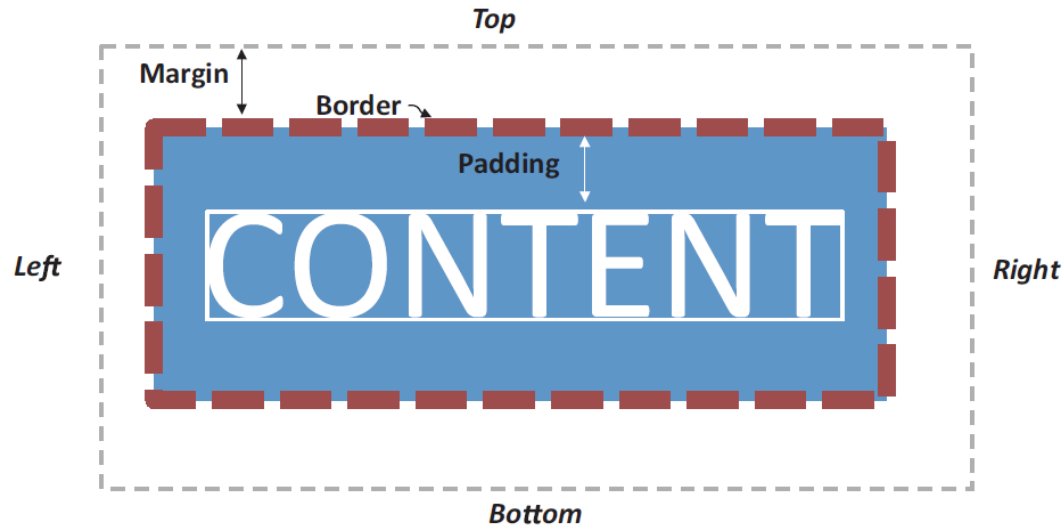


Figure 4-9

# CSS Box Model (continued)

- Each content box can have margins, borders, and padding
- The **margin** provides passive white space between block elements or between the top or bottom of a webpage
- The **border** separates the padding and the margin of the block element
- **Padding** is the passive white space between the content and the border of a block element

# CSS Box Model (continued 1)

**Table 4–4 Common CSS Box Model Properties**

Property	Description	Examples
margin	Sets the amount of space around the block element (top, right, bottom, left)	margin: 20px; margin-top: 2em; margin-bottom: 150%;
padding	Sets the amount of space between content and the border of its block element	padding: 10px; padding-left: 1.5em; padding-right: 125%;
border	Sets the format of the block element's border	border: solid 1px #000000;
border-style	Designates the style of a border	border-top-style: solid; border-top-style: dotted;
border-width	Designates the width of a border	border-top-width: 1px; border-bottom-width: thick;
border-color	Designates the border color	border-top-color: #000000; border-bottom-color: gray;
border-radius	Rounds the corners of a block element's border	border-radius: 10px;
box-shadow	Adds a shadow to a block element's border	box-shadow: 8px 8px 8px #000000;



# Selectors

- A style rule begins with a selector, which specifies the element to style
- A selector can be
  - an HTML element name
  - an id attribute value
  - a class attribute value
- An **id** or a **class selector** is used to apply styles to p elements

# Selectors (continued)

- An **id selector** uses the id attribute value of an HTML element to select a single element
- For example, to style the `div id="container"` element, use `#container` as the selector

```
#container {  
    border: solid 2px;  
}
```

# Selectors (continued 1)

- A **class selector** is used to select elements that include a certain class attribute
- For example, to style `class="mobile"`, use `.mobile` as the selector

```
.mobile {  
    font-size: 10pt;  
}
```

# Selectors (continued 2)

- A **descendant selector** is used to create style that applies to an element contained within another element
- For example, the following style rule sets the list-style property to none for list items in an unordered list included in the navigation area:

```
nav ul li {  
    list-style: none;  
}
```

# To Create a CSS File and a Style Rule for the Body Element

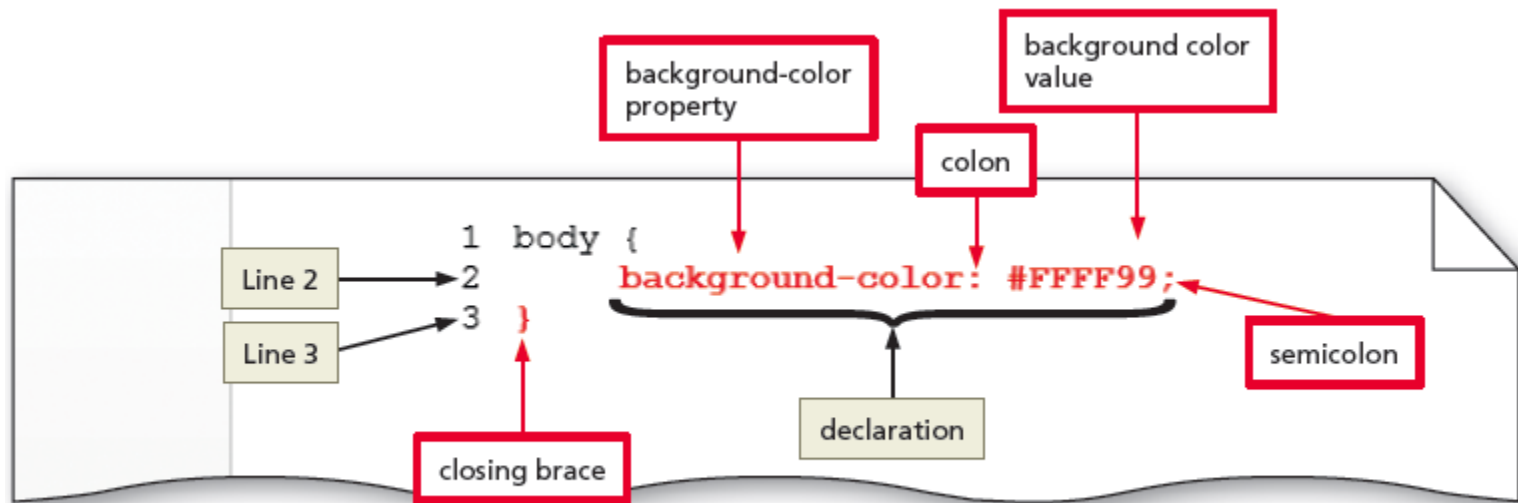


Figure 4-12

# Linking an HTML Document to a CSS File

- After creating a CSS file, link it to all the webpages that will use its styles
- Insert a `link` element on the HTML page within the `<head>` and `</head>` tags
- The `link` element uses two attributes:
  - `rel`
  - `href`
- The `rel` attribute uses the `stylesheet` value to indicate that the document is linked to a style sheet

# Linking an HTML Document to a CSS File (continued)

- The `href` attribute value specifies the file path or file name of the CSS file
- Following is an example of a link to a style sheet named `styles.css` and stored in the `css` folder:

```
<link rel="stylesheet"  
href="css/styles.css">
```

- The `type="text/css"` attribute and value is also commonly used within a `link` element to reference a CSS file

# Aligning Webpage Content

- One way to align webpage content is to use the text-align property, which applies to block elements
- The text-align property can use left (the default), center, right, or justify as its value
- For example, the following rule centers an h1 element:

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



# To Center Content

- To center all of the elements of a webpage using a single style rule, set the left and right margins to **auto**
- In addition, set the width to 80% so that the elements do not span 100 percent of the browser window

# Creating Style Rules for Structural Elements

- The header section appears at the top of a webpage and thus needs formatting that makes the header contents stand out and attract visitors to the page
- The nav section should be formatted differently from the other structural elements as it should be prominent and easy to find on the webpage
- The main section should be formatted using the display property

# Creating Style Rules for Structural Elements (continued)

- To apply text and box model properties to the main section and have them appear as intended, the **display property** is used
- Create a style rule that formats the footer section by defining the font size, text alignment, and top margin of the **footer** element

# To Create a Style Rule for the Header Element

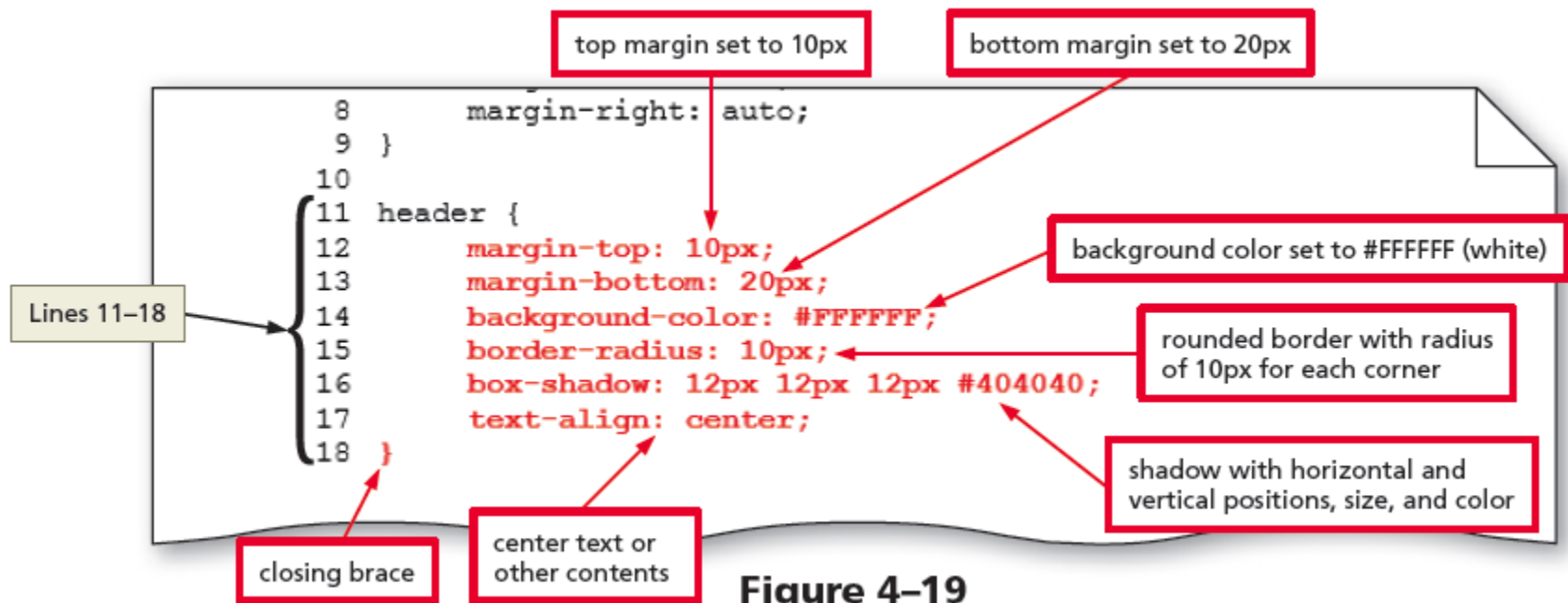


Figure 4-19

# To Create a Style Rule for the Nav Element

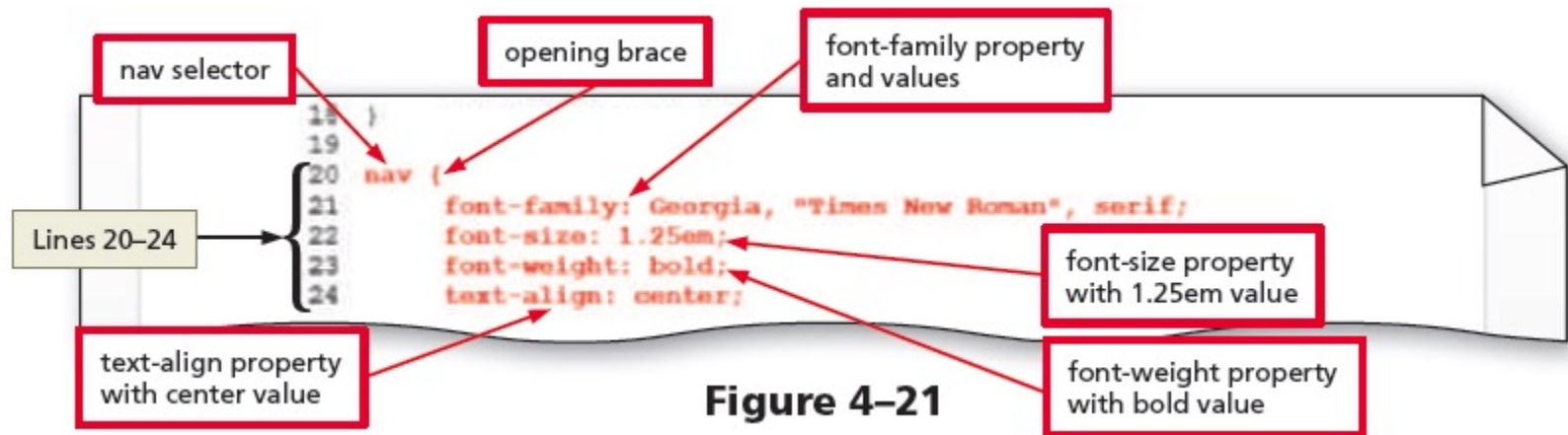


Figure 4-21

# To Create a Style Rule for the Main Element

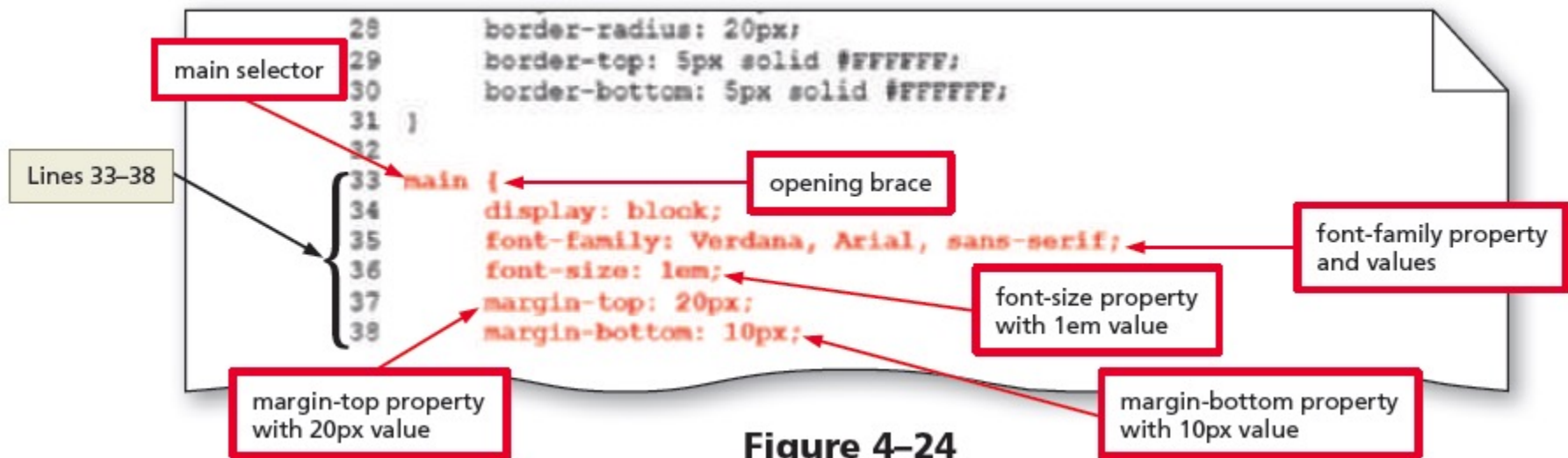


Figure 4–24

# To Create a Style Rule for the Footer Element

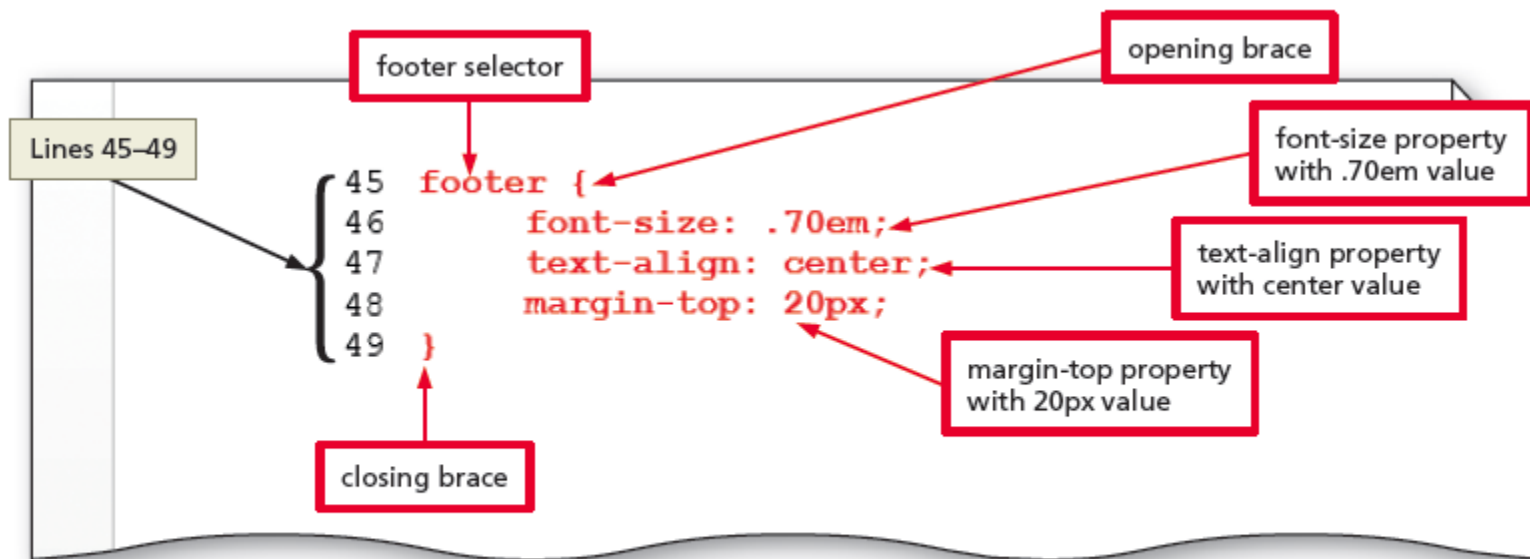


Figure 4-27

# Creating Style Rules for Classes

- Consider the following example:

```

```

- The `img` element displays the `equipment1.jpg` image
- The first attribute and value, `class="equip"`, assigns this element to the `equip` class
- Including the `class="equip"` attribute and value in each `img` element helps format all the elements assigned to the `equip` class with a single style rule



# Creating Style Rules for Classes (continued)

- For example, the following style rule adds 20 pixels of padding to the right side of elements in the `equip` class:

```
.equip {  
    padding-right: 20px;  
}
```

# Creating Style Rules for Classes

## (continued 1)

- To indicate a class name as a selector, include a period (.) before the class name
- **Float property** – It positions an element to the right or left of other elements
- **Clear property** – It removes the float effect from a webpage

# Using CSS List Properties

- The CSS **list-style properties** are used to control the appearance of numbered and bulleted lists
- Lists marked with the `<ul>` and `</ul>` tags display a solid bullet before each list item
- Lists marked with the `<ol>` and `</ol>` tags display Arabic numerals (1, 2, 3, and so on) before the list items
- For example,

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

# Using CSS List Properties (continued)

- The default value for the `list-style-position` property is `outside`, which displays the list item with a bullet or number `outside` of the list's content block as in the following text:
  1. Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

# Using CSS List Properties (continued 2)

- Using `inside` as the value displays the bullet or number inside the list's content block, as in the following text:
  - Morbi odio nisl, facilisis non egestas a, tristique vitae neque.
  - Lorem ipsum dolor sit amet, consectetur adipiscing elit.

# Adding Comments to CSS Files

- Comments provide additional information about the area where the styles are applied or other helpful explanations, such as what the styles do
- The syntax for a comment is as follows:  

```
/* Place your comment here */
```

# To Validate the CSS File

- The following steps validate a CSS file
  - Open the browser and type `http://jigsaw.w3.org/css-validator/` in the address bar to display the W3C CSS Validation Service page
  - Tap or click the By file upload tab to display the Validate by file upload information
  - Tap or click the Browse button to display the Choose File to Upload dialog box
  - Navigate to your css folder to find the styles.css file (Figure 4–41)

# To Validate the CSS File (continued)

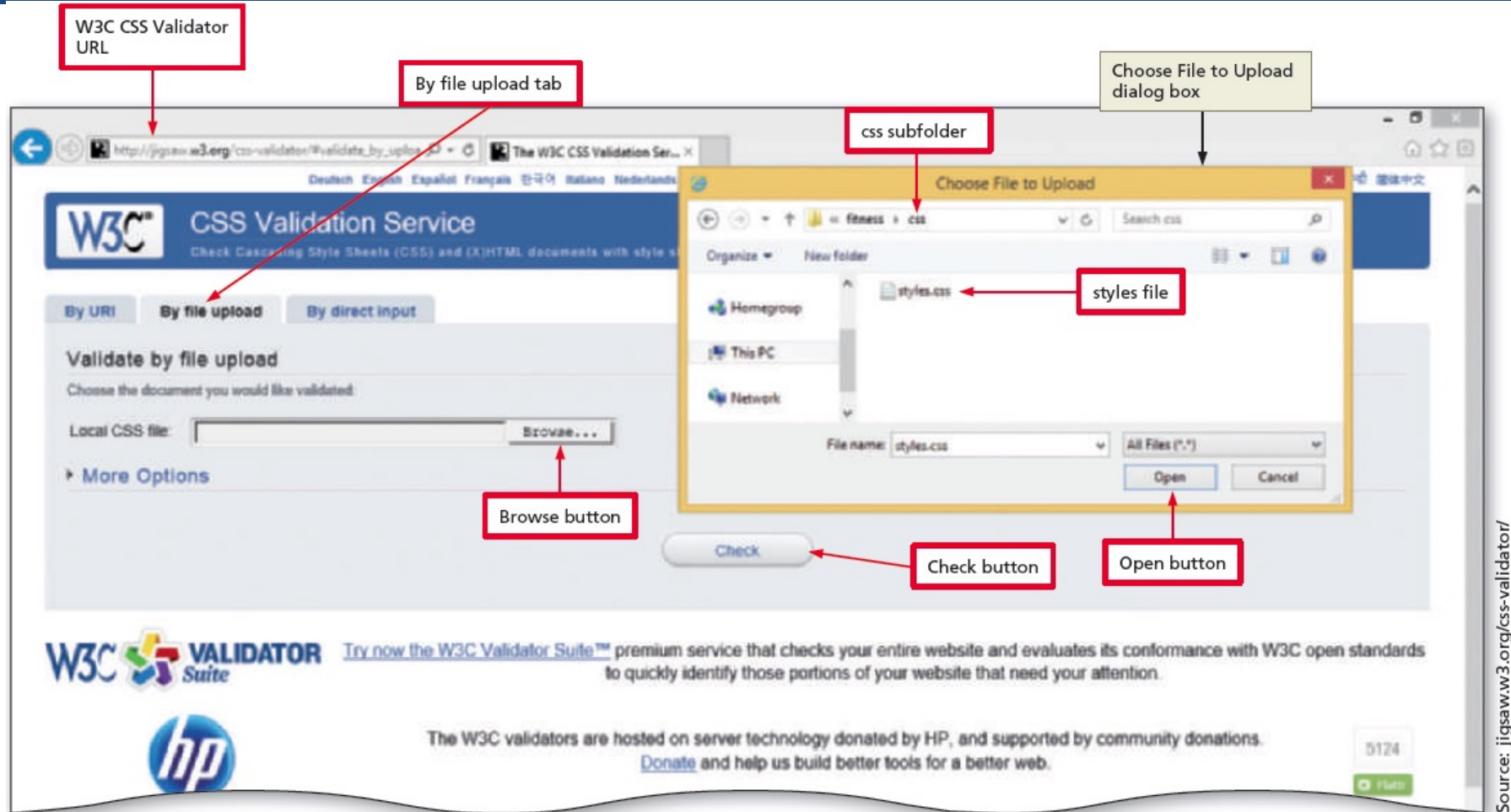


Figure 4-41



# To Validate the CSS File (continued 1)

- Tap or click the styles.css document to select it
- Tap or click the Open button to upload the selected file to the W3C CSS validator
- Tap or click the Check button to send the document through the validator and display the validation results page (Figure 4–42)

# To Validate the CSS File (continued 2)

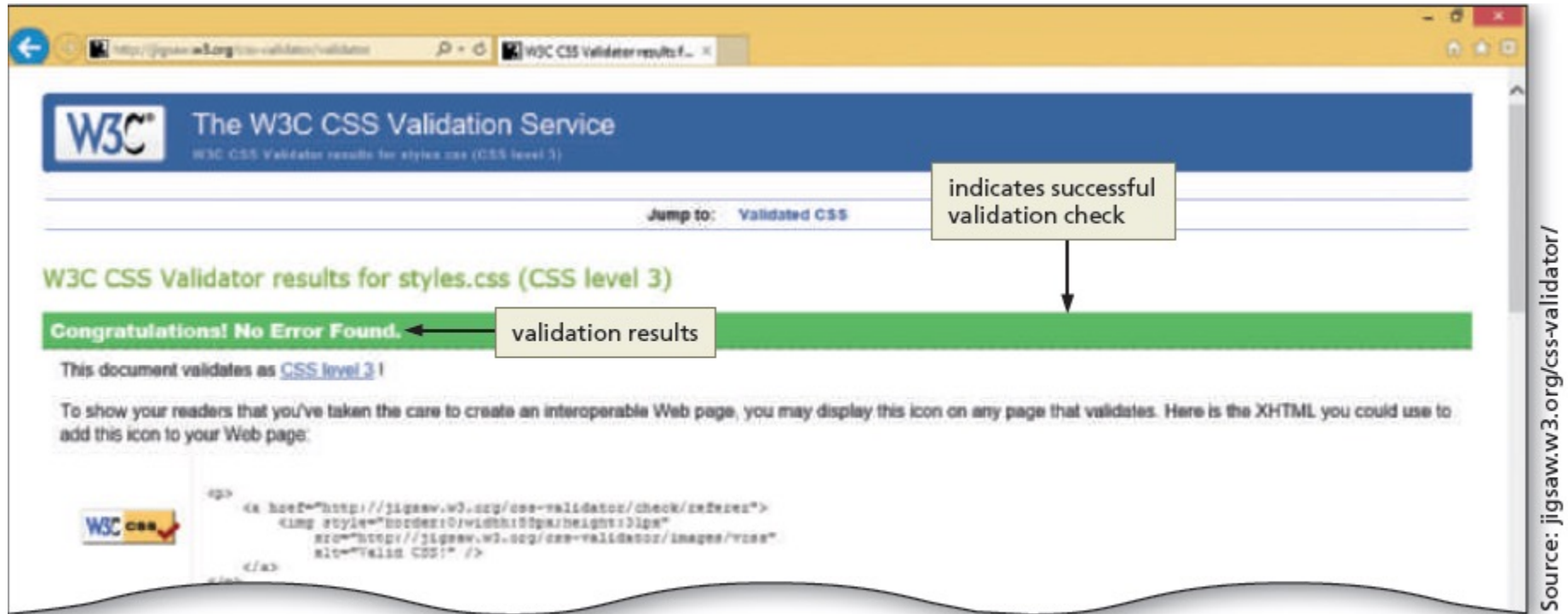


Figure 4-42

Source: [jigsaw.w3.org/css-validator/](http://jigsaw.w3.org/css-validator/)

# HTML

## 8<sup>th</sup> Edition

### Chapter 4 Complete

