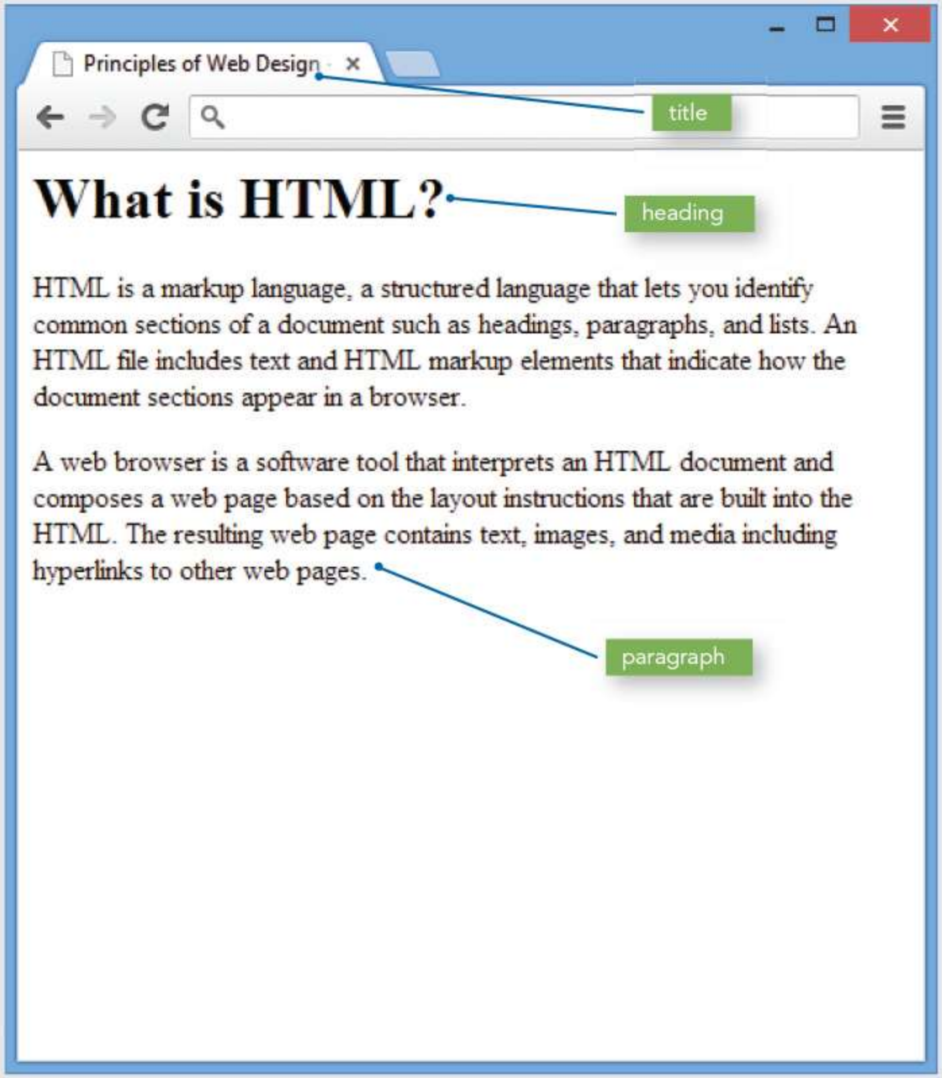
Client-Side Cascade Style Sheets

By: Joel Sklar

Saja Al-Mamoori

HTML in the Browser

- The browser interprets the HTML markup elements and displays the results, hiding the actual markup from the user
- Each browser interprets HTML in its own way, based on its rendering engine
- It is essential that you test your work in different web browsers



What is HTML?

HTML is a markup language, a structured language that lets you identify common sections of a document such as headings, paragraphs, and lists. An HTML file includes text and HTML markup elements that indicate how the document sections appear in a browser.

A web browser is a software tool that interprets an HTML document and composes a web page based on the layout instructions that are built into the HTML. The resulting web page contains text, images, and media including hyperlinks to other web pages.

Figure 1-7: Browser interprets the HTML markup and displays the results

Microsoft product screenshots used with permission from Microsoft Corporation.

Adding Style with CSS

- Web designers use Cascading Style Sheets (CSS) to add presentation information to web pages
- With CSS you can display information for different devices
- With style sheets, the presentation properties are separate from the content
- CSS lets you control the presentation characteristics of an entire web site with a single style sheet

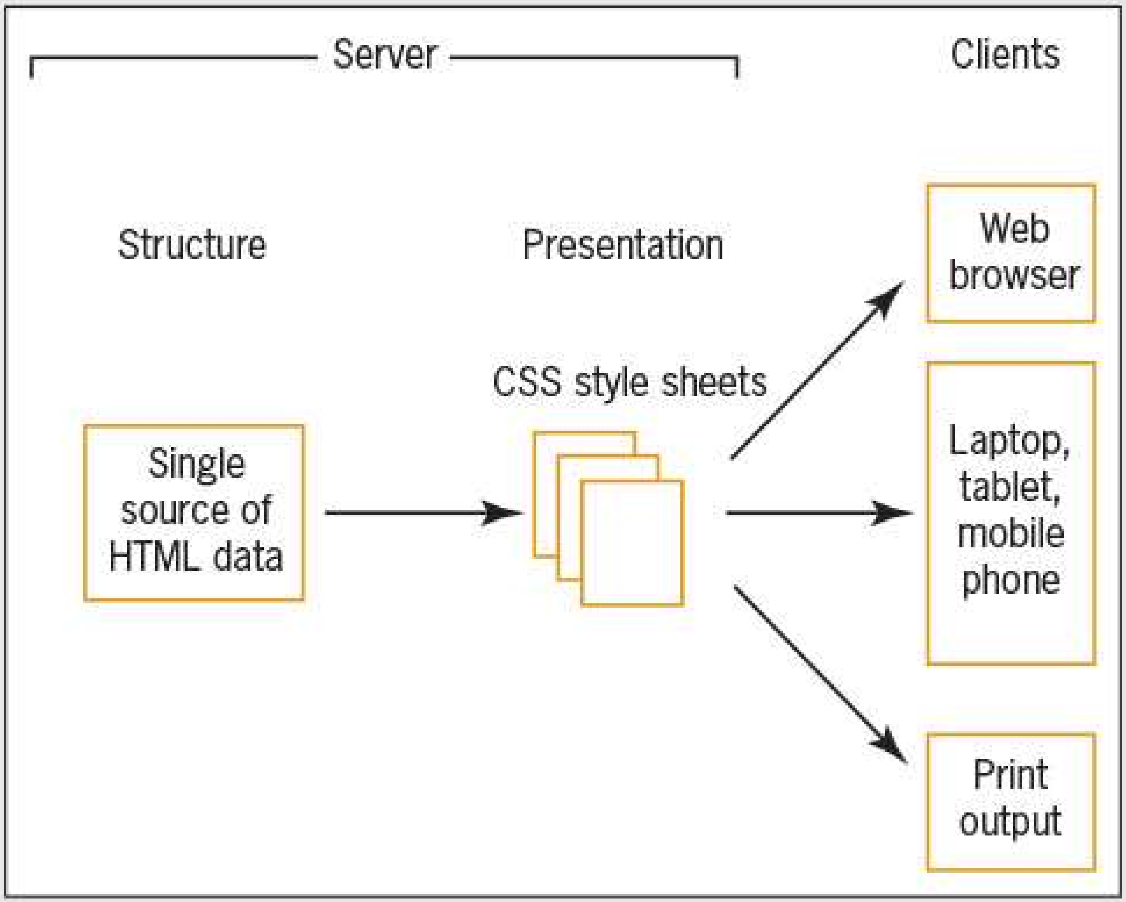
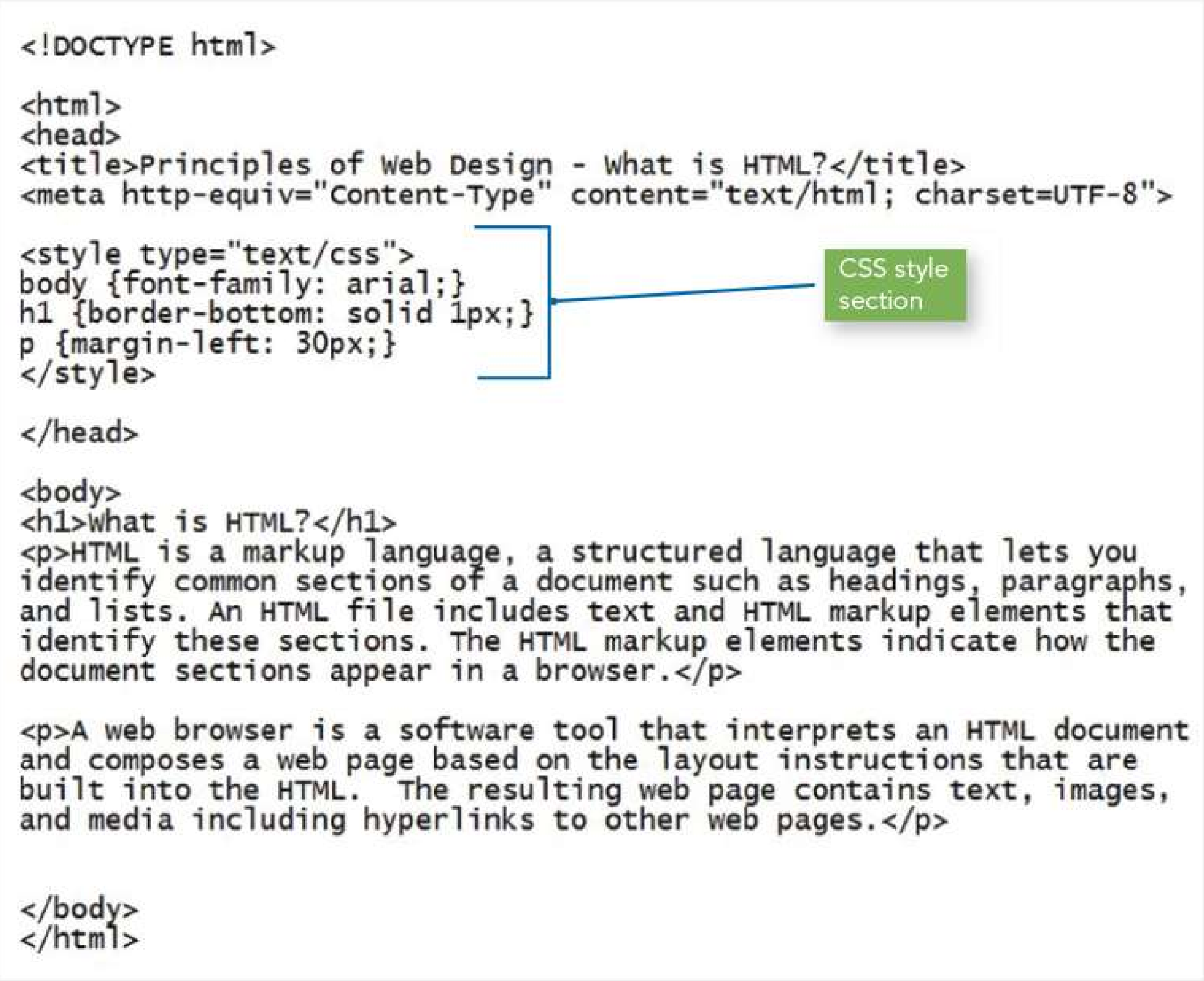


Figure 1-8: Formatting data for multiple destinations

© 2015 Cengage Learning®

Adding Style with CSS

- The next two figures show CSS style rules and the result in the browser
- The style rules in Figure 1-9 specify that the body text for the page will be Arial, the h1 will have a bottom border, and the paragraph will have a 30-pixel left margin
- Figure 1-10 shows the results in the browser



```
<!DOCTYPE html>

<html>
<head>
<title>Principles of Web Design - What is HTML?</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<style type="text/css">
body {font-family: arial;}
h1 {border-bottom: solid 1px;}
p {margin-left: 30px;}
</style>

</head>

<body>
<h1>What is HTML?</h1>
<p>HTML is a markup language, a structured language that lets you
identify common sections of a document such as headings, paragraphs,
and lists. An HTML file includes text and HTML markup elements that
identify these sections. The HTML markup elements indicate how the
document sections appear in a browser.</p>

<p>A web browser is a software tool that interprets an HTML document
and composes a web page based on the layout instructions that are
built into the HTML.  The resulting web page contains text, images,
and media including hyperlinks to other web pages.</p>


</body>
</html>
```

Figure 1-9: CSS style section contains presentation information

CSS Syntax

- Style rules express style characteristics for an HTML element
- For example, the following style rule sets all <p> elements to blue text:

```
P {color: blue;}
```

- Style rules contain a selector and a declaration
- You will learn more about CSS in Chapter 4

The Evolution of CSS

- CSS was developed to **standardize display information**
- CSS was slow to be supported by browsers
- Newer browsers offer more complete support
- Latest release is **CSS3**
- In CSS, **style rules** express the style characteristics for an HTML element
- A set of style rules is called a **style sheet**
- Style rules are easy to write and interpret

CSS Style Rules

- Style rules are composed of two parts: a **selector** and a **declaration**
- The selector determines the element to which the rule is applied
- The declaration details the exact property values

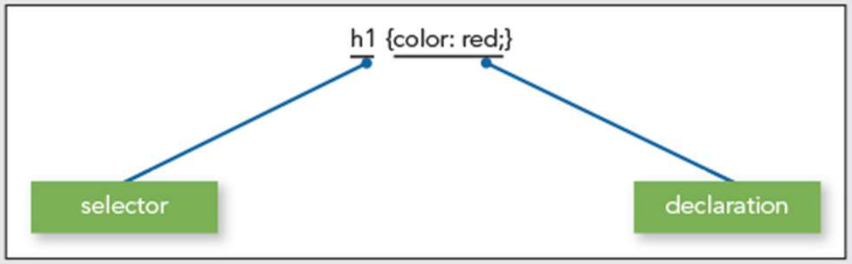


Figure 4-1: Style rule syntax

© 2015 Cengage Learning®

CSS Style Rules

- The declaration contains a **property** and a **value**
- The property is a quality or characteristic
- The precise specification of the property is contained in the value
- CSS includes a wide variety of different properties, each with a specific number of values

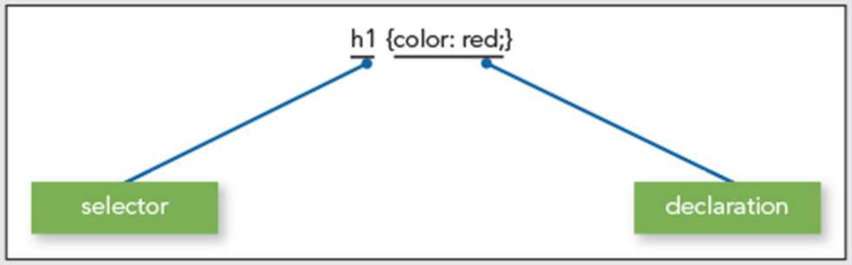
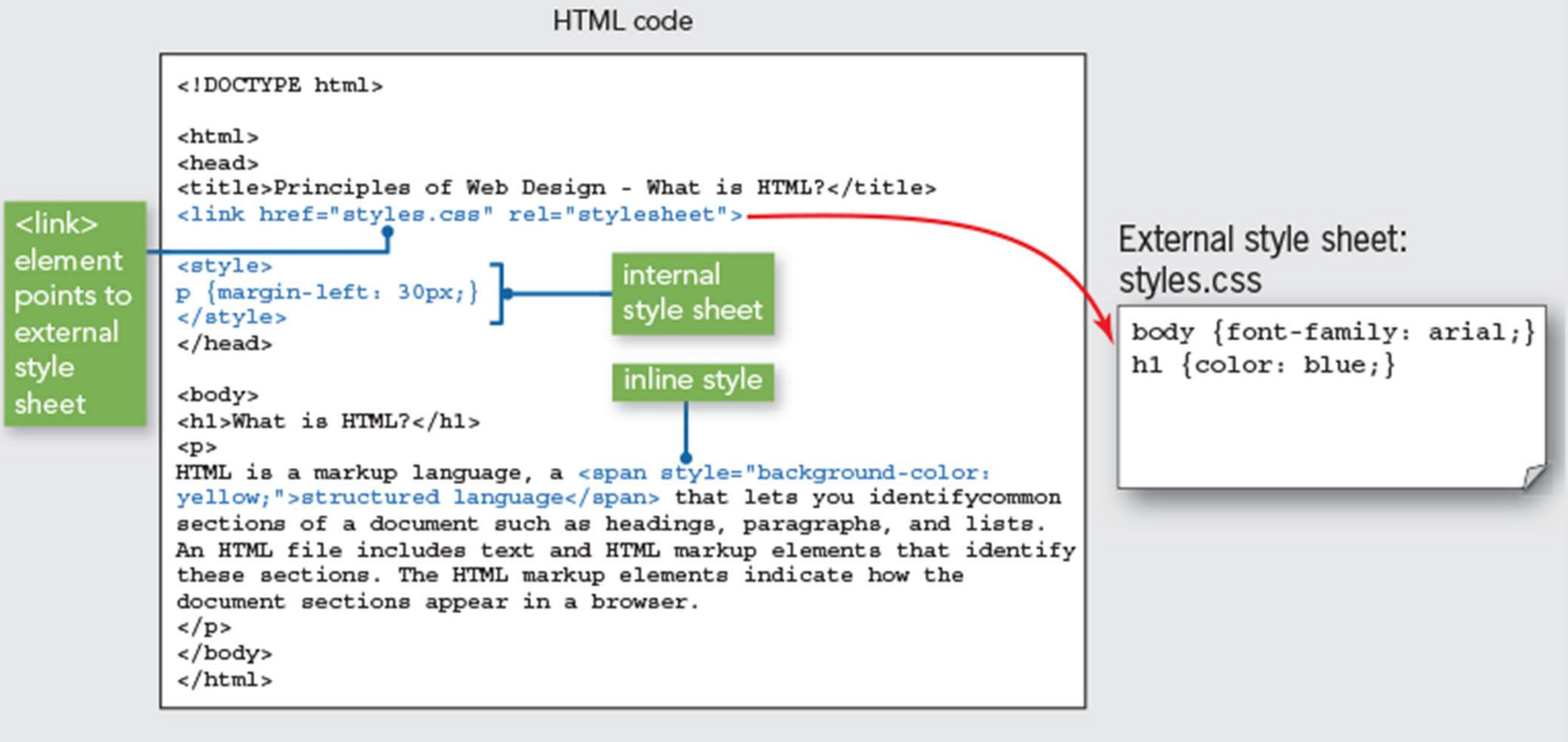


Figure 4-1: Style rule syntax

© 2015 Cengage Learning®


HTML code
<link> element points to external style sheet
internal style sheet
inline style
External style sheet: styles.css

```
<!DOCTYPE html>

<html>
<head>
<title>Principles of Web Design - What is HTML?</title>
<link href="styles.css" rel="stylesheet">
<style>
p {margin-left: 30px;}
</style>
</head>

<body>
<h1>What is HTML?</h1>
<p>
HTML is a markup language, a <span style="background-color:
yellow;">structured language</span> that lets you identifycommon
sections of a document such as headings, paragraphs, and lists.
An HTML file includes text and HTML markup elements that identify
these sections. The HTML markup elements indicate how the
document sections appear in a browser.
</p>
</body>
</html>
```

```
body {font-family: arial;}
h1 {color: blue;}
```

Figure 4-3: Three methods of combining CSS with HTML

© 2015 Cengage Learning®

Using External Style Sheets

- External style sheets let you specify rules for **multiple web pages,** text documents that contain style rules
- External style sheets have a .css extension
- The **link element** lets you specify an external style sheet
- It is used within the <head> section of a document

```
<head>
<title>Sample Document</title>
<link href="styles.css" rel="stylesheet"
type="text/css">
</head>
```

Using Internal Style Sheets

- Internal style sheets are contained within the **<style>** element
- The style element is contained within the <head> section of the document
- Style rules contained in an internal style sheet only affect the document in which they reside

```
<head>
<title>Sample Document</title>
<style>
h1 {color: red;}
</style>
</head>
```

Using Inline Styles

- You can define styles for a single element with the style attribute
- The **style attribute** can be used to override a style that was set at a higher level
- The style attribute is useful for testing styles during development
- This is the least used method of applying CSS styles

```
<h1 style="color: blue">Some Text</h1>
```

Using Inheritance to Write Simpler Style Rules

- Elements in an HTML document are **structured in a hierarchy**
- Parent elements contain child elements
- Elements can be both parent and child elements
- The CSS properties **inherit** from parent to child
- The property descriptions list whether a property is inherited
- You can style multiple document elements with just a few style rules using inheritance

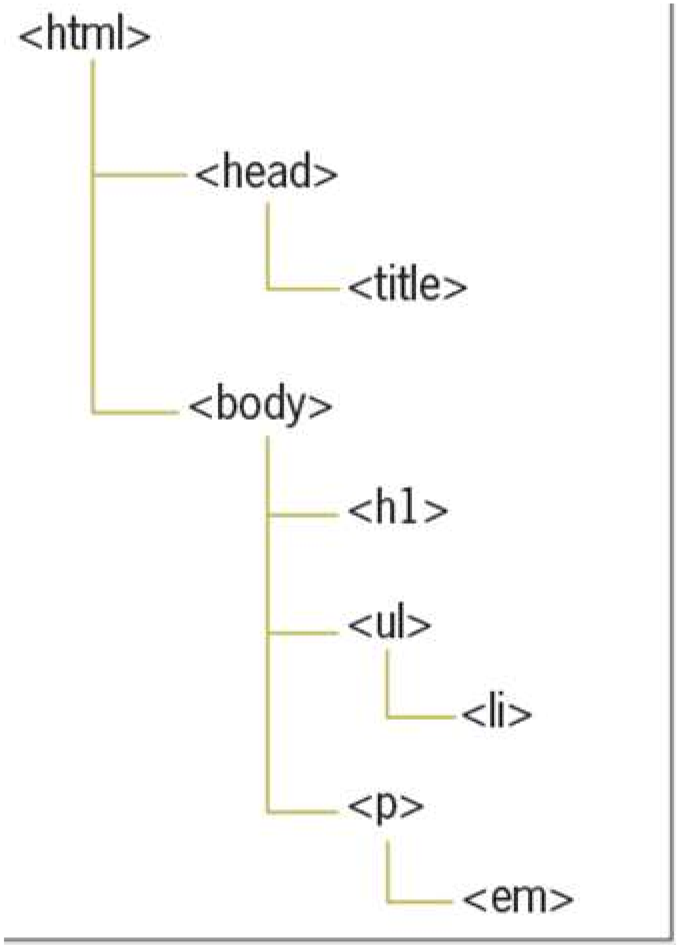
Using Inheritance to Write Simpler Style Rules

- Specific style rules:

  ```
  <style>
  h1 {color: red;}
  p {color: red;}
  ul {color: red;}
  em {color: red;}
  li {color: red;}
  </style>
  ```

- Using inheritance:

  ```
  <style>
  body {color: red;}
  </style>
  ```

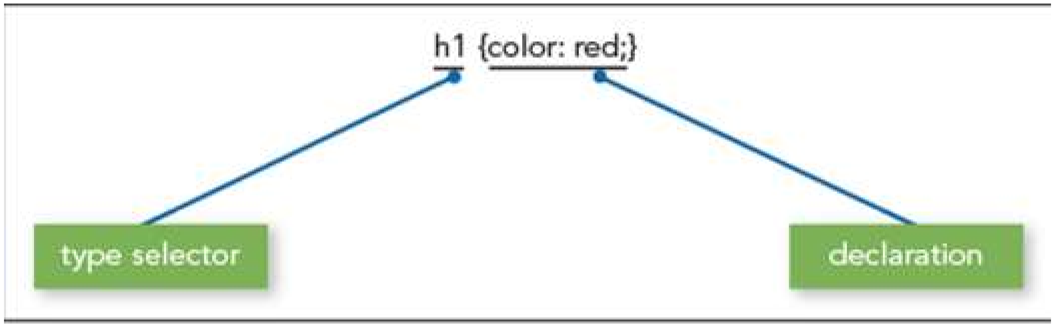


Examining Basic Selection Techniques

- In this section, you will review style rule syntax and learn about the following basic selection techniques:
  - Using type selectors
  - Grouping selectors
  - Combining declarations
  - Using descendant selectors

Using Type Selectors

- The **selector** determines the element to which a style declaration is applied
- They select every element in the document that matches the style rule



Grouping Selectors

- You can group selectors to which the same rules apply
- Specific style rules:

```
h1 {color: red;}
h2 {color: red;}
```

- Grouping selectors:

```
h1, h2 {color: red;}
```

Combining Declarations

- You can state multiple property declarations for the same selector
- Specific style rules:

```
p {color: blue;}
p {font-size: 125%;}
```

- Combining declarations:

```
p {
color: blue;
font-size: 125%;
}
```

Using Descendant Selectors

- You can select elements that are descendents of other elements
- Selecting only <em> elements that are contained within <p> elements

```
p em {color: blue;}
```

Using the Universal Selector

- Universal selector lets you select groups of elements
- Selecting all children of the div element:

```
div * {font-family: sans-
serif;}
```

  - * Means any element but not Zero

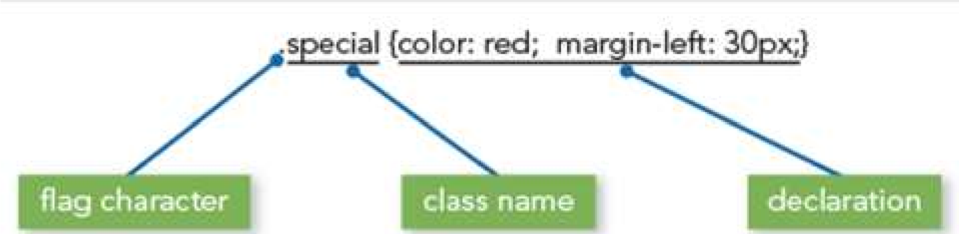
div * em { color:red}

```
<body>
 <div> <h1>The <em>Universal</em>
           Selector</h1>
<p>We must
 <em>emphasize</em>
 the following:</p>
<ul> <li>It's <em>not</em>
a wildcard.</li>
 <li>It matches elements regardless of
 <em>type</em>.
</li>
</ul>
This is an <em>immediate</em>
child of the division.
</div>
</body>
```

https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default

Using the class Selector

- The class selector lets you **write rules and give them a name**
- You can apply that name to **any element** you choose, using class attribute of the element
- The period (.) flag character indicates the selector is a class selector



```
.special {color: red;  margin-left: 30px;}
```

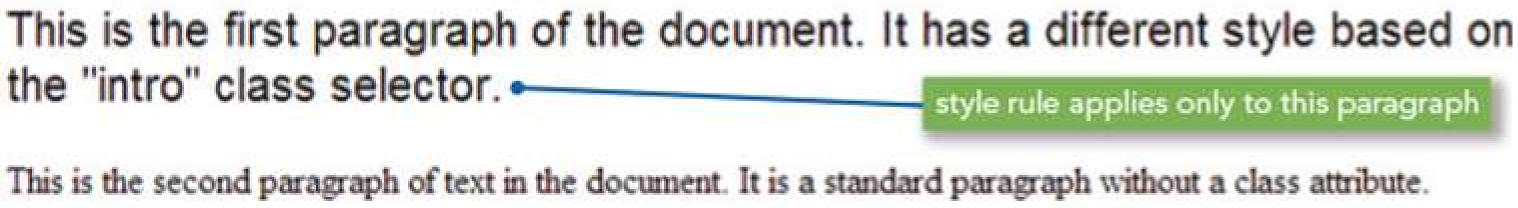
Using the Class Selector

- Style rule:

```
.intro {font-size: 125%;}
```

- Applied in document:

```
<p class="intro">This is the first
paragraph…</p>
```



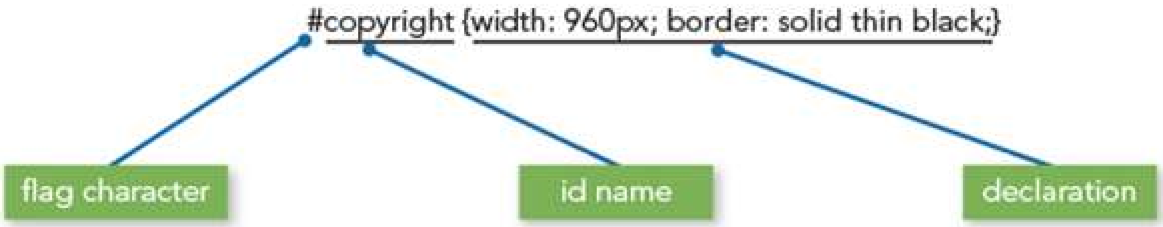
This is the first paragraph of the document. It has a different style based on the "intro" class selector.

This is the second paragraph of text in the document. It is a standard paragraph without a class attribute.

Using the id Selector

- The difference between id and class is that id refers to only **one instance** of the id attribute value within a document
- The ID attribute is used to identify layout sections of the page
- The ID attribute uses a pound sign (#) flag character

Using the id Selector



```
#copyright {width: 960px; border: solid thin black;}
```

- Applied in document:

```
<p id="copyright">This is the
copyright information for the
page.</p>
```

Using the <div> and <span> Elements

- The <div> (division) and <span> (span of words) elements are designed to be used with CSS
- They let you specify logical divisions within a document that have their own name and style properties

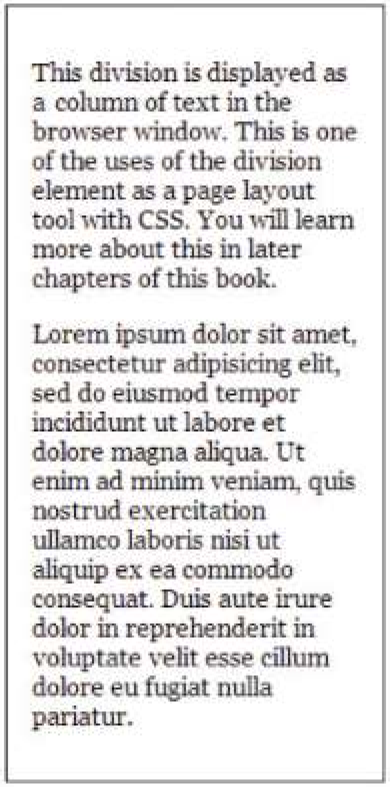
Working with <div> elements

- Use <div> with the class and ID attributes to create logical divisions on a web page
- A division with an id named column as the selector:

```
div#column {
width: 200px;
height: auto;
padding: 15px;
border: thin solid;
}
```

Applied in the document:

```
<div id="column">This division
displays… </div>
```



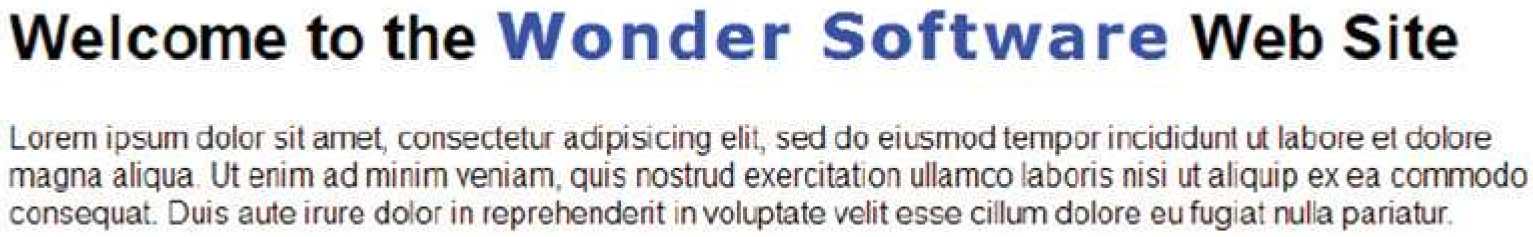
This division is displayed as a column of text in the browser window. This is one of the uses of the division element as a page layout tool with CSS. You will learn more about this in later chapters of this book.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Try it yourself: http://www.w3schools.com/code/tryit.asp?filename=FBQ1MES32VSI

Working with <span> elements

- The span element lets you specify **in-line elements** that have their **own name** and **style properties**
- In-line elements reside within a line of text



Welcome to the Wonder Software Web Site

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.

Working with <span> elements

- Style rule:

```
span.logo {
color: white;
background-color: black;
}
```

- Applied in document:

```
<p>Welcome to the <span
class="logo">Wonder
Software</span>Web site.</p>
```

http://www.w3schools.com/code/tryit.asp?filename=FBQZ7L8X3P61

Using Attribute Selectors

- Attribute selectors let you select an element based on whether the element contains an attribute
- Elements can be selected based on a specific value the attribute contains

Using Attribute Selectors

- Attribute selectors match against attributes and their values
- To select this element:

```
<img src="images/home.gif"
title="home" alt="Home navigation
button" />
```

using attribute selectors, you could use the value that the title attribute contains, as shown:

```
img[title=home] {border-color: red;}
```

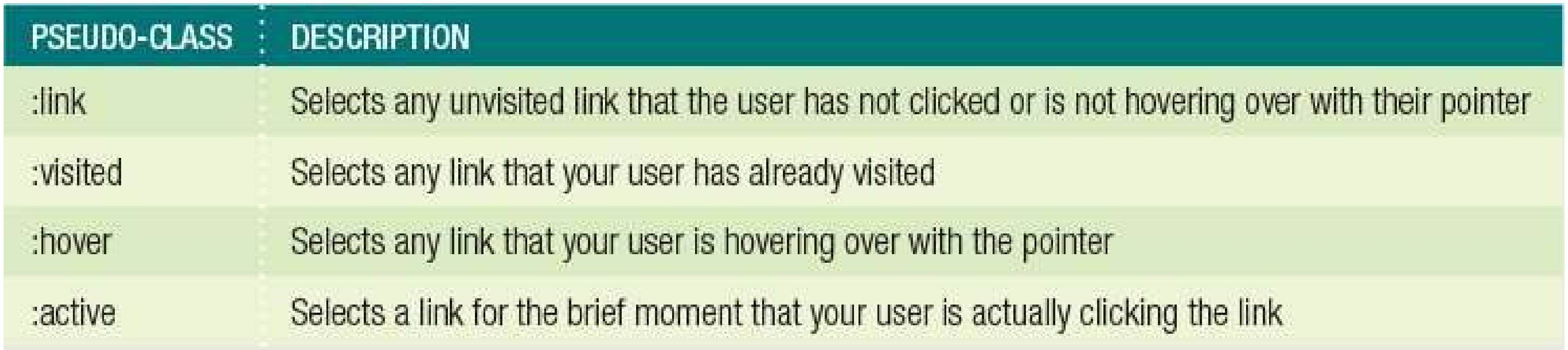
Using Pseudo-Class and Pseudo-Element Selectors

- Pseudo-classes select elements based on characteristics other than their element name
- For example, you can change the characteristics of hypertext links with pseudo-classes
- Pseudo-elements let you change other aspects of a document that are not classified by standard elements such as the first letter or line of a paragraph

Using the Link Pseudo-Classes

- The **link pseudo-classes** let you change the **style characteristics** for four different hypertext link states
- These pseudo-classes only apply to the <a> element with an href attribute

Using the Link Pseudo-Classes



PSEUDO-CLASS	DESCRIPTION
:link	Selects any unvisited link that the user has not clicked or is not hovering over with their pointer
:visited	Selects any link that your user has already visited
:hover	Selects any link that your user is hovering over with the pointer
:active	Selects a link for the brief moment that your user is actually clicking the link

Table 4-3: Link Pseudo-Classes

Using the Link Pseudo-Classes

- Because of the specificity of the pseudo-class selectors, you should always place your link pseudo-class in the following order:

  1. Link
  2. Visited
  3. Hover
  4. Active

Using the Link Pseudo-Classes

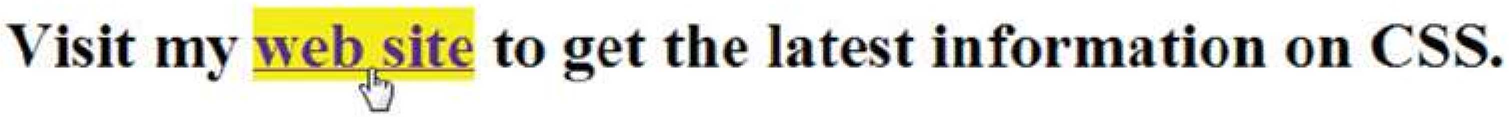
- The following rules change the colors of the hypertext links:

```
:link {color: red;}
:visited {color: green;}
```

Using the :hover Pseudo-Class

- The :hover pseudo-class lets you apply a style that appears when the user points to an element with a pointing device

```
a:hover {background-color:
yellow;}
```



Visit my web site to get the latest information on CSS.

Using the :before and :after Pseudo-Elements

- These psuedo-elements let you insert created content
- These are useful for repeated content
- For example, the following style rule inserts the word *Figure* followed by a colon before an <P> text that has the class figtitle:

```
p.figtitle:before {content: “Figure: ”;}
```

Understanding How the Cascade Affects Style Rules

- The cascade means that multiple style sheets and style rules can apply to the same document
- Only one rule can apply to an element
- The CSS cascading mechanism determines which rules apply based on three variables:
  - Specificity of the selector
  - Order of the rule in the style sheet
  - Use of the !important keyword

Determining Rule Weight by Specificity

- If we have:

```
p {color: red;}
div p {color: blue;}
```

- The first applies to all <p> elements on the page.
- the second applies to <p> elements that are specifically within a <div> element in the page.
- check it out please:
- http://www.w3schools.com/code/tryit.asp?filename=FBPUL69TIZAB

Determining Rule Weight by Order

- Rules that are included later in the style sheet order take precedence over earlier rules.

Check it out:

http://www.w3schools.com/code/tryit.asp?filename=FBPUQE50VG8Y

Determining Rule Weight with the !important Keyword

- The !important specifies that a rule should take precedence no matter what order the rules in the style sheet.
- Check it out:
- http://www.w3schools.com/code/tryit.asp?filename=FBPUVOS7HF2Z