Chapter 4: Cascading Style Sheets

Objectives

- recognize benefits of CSS
- build a basic style sheet
- use inheritance to write simpler style rules
- examine & apply basic selection techniques
- use class, id, and other selectors
- use `<div>` and `<span>` elements

CSS

CSS was developed to standardized display information. Latest release is CSS3. CSS can be combined with HTML through *inline*, *internal*, and *external* style sheets.

Style Rules

- style rules express the style characteristics for an HTML element
- set of style rules is a style sheet
- composed of two parts: **selector** and **declaration**
- selector determines the element to which the rule is applied
- declaration details the exact property values

External Style Sheets

- external sheets let you specify rules for multiple web pages
- text documents (.css extension) that contain style rules
- usage: `<head><link href='style.css' rel='stylesheet' type='text/css'></head>`

Internal Style Sheets

- contained w/in the `<style>` element, which is contained in the `<head>` section

  ```
  <head><style> ... </style></head>
  ```

Inline Styles

- define styles for a single attribute using the style attribute
- can override a style that was set at a higher level

  ```
  <h1 style='color: blue'>text</h1>
  ```

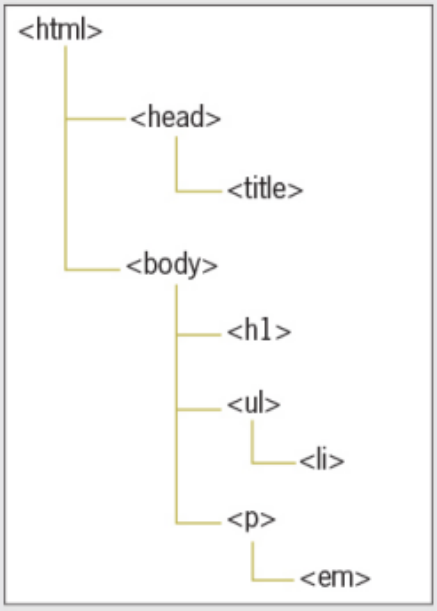

Figure 4-6: HTML document structure

Using Inheritance to Write Simpler Style Rules

- HTML elements are structured in a hierarchy
- parent elements contain child elements. An element can be both parent and child elements
- CSS properties inherit from parent to child
- property descriptions list whether a propert is inherited

Basic Selection Techniques

Type Selectors

- selector determines the element to which a style declaration is applied
- type selectors select every element in the doc that matches the style rule
- declarations can be combined

Figure 4-7: Type selector syntax

Grouping Selectors

- group selectors are similar to type selectors, but with a list of types separated by a comma

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

Descendant Selectors

- you can select elements that are descendants of other elements

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

Universal Selector

- allows selection of groups of elements

Example: select all children of the div element

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

Class Selector

- select elements of an HTML doc using: `class`, `ID`, `<div>`, `<span>`, `pseudo-class`, `pseudo-element`
- class allows for writing a set of rules and giving them a name. The rule name is applied to elements
- the period (.) flag indicated the selector is a class selector

ID Selector

- id refers to only one instance of the id value within a document
- used to identify layout sections of the page
- uses a pound sign (#) flag character

`<div>` and `<span>` elements

- specify logical divisions w/in a document that have their own name and style properties
- use `<div>` with the class and ID attributes to create logical divisions

```
div#column { width: 200px; height: auto; }
```

- `<span>` lets you specify inline elements that have their own name and style properties
- **inline elements** reside within a line of text
- span is applied the same way div is

Attribute Selectors

- select an element based on whether the element contains an attribute
- select an element based on a specific value the attribute contains

Usage: `<img src='...' title='home' alt='home nav button'>` `img[title=home]{border-color: red;}`

Pseudo-Class Selector

- pseudo-classes select elements based on characteristics other than their element name
- lets you change style characteristics for four different hypertext link states
- only apply to the `<a>` element with an href attribute
- always place link pseudo-class in the order they appear in the following table

Pseudo-Class	Description
:link	Selects any unvisited link that the user has not clicked or is not hovering over
:visited	selects any link that the user has already visited
:hover	selects any link that the user is hovering over with the pointer
:active	Selects a link for the brief moment that the user clicks the link

Pseudo-Element Selector

- pseudo-elements change other aspects of a doc that are not classified by standard elements such as the first letter or line of a paragraph
- use the `:first-letter` pseudo-element to apply style rules to the first letter of an element
- use the `:first-line` pseudo-element to apply syle rules to the first line of text in an element
- use `:before` and `:after` to insert content. Useful for repeated content
  - example: this style rule inserts the word `Figure:` before a p.figtitle element `p.figtitle:before{content: 'Figure: ';}`

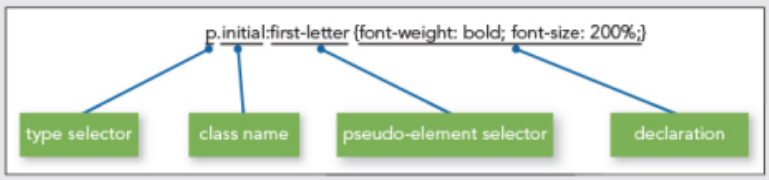

Figure 4-16: Using a class selector with a :first-letter pseudo-element

How Cascade Affects Style Rules

- cascade means multiple style sheets and rules can apply to the same document
- only one rule can apply to an element
- CSS cascading mechanism determines which rules apply based on 3 variables:
  - specificity of the selector
  - order of the rule in the style sheet
  - use of the `!important` keyword

Determining Rule Weight by Specificity

- rules included later in the style sheet order take precedence over earlier rules
- `!important` specifies a rule should take precedence no matter what order

Summary

- CSS rules can be combined with HTML code in a number of ways
- CSS uses inheritance and cascading to determine which style rules take precedence
- selectors and declarations can be combined in many ways
- many ways to select elements
- `class` & `ID` attribute selectors are often paired with `<div` and `<span>` elements
- pseudo-class and pseudo-element selectors let you change color and styling of links and other elements of a document