3.1 Introduction

- HTML is primarily concerned with content, rather than style
  - However, tags have presentation properties, for which browsers have default values
- The CSS1 cascading style sheet specification was developed in 1996; followed by CSS2 in 1998
  - Provide the means to control and change presentation of HTML documents
  - Not technically HTML, but can be embedded in HTML documents
  - One of the most important capabilities of style sheets is that they allow you to impose a standard style on a whole document, or even a whole collection of documents
- Style is specified for a tag by the values of its properties
- Current browsers do not implement all of CSS2

3.2 Levels of Style Sheets

- There are three levels of style sheets
  - Inline - specified for a specific occurrence of a tag and apply only to that tag
    - This is fine-grain style, which defeats the purpose of style sheets - uniform style
  - Document-level style sheets - apply to the whole document in which they appear
  - External style sheets - can be applied to any number of documents
- When more than one style sheet applies to a specific tag in a document, the lowest level style sheet has precedence
  - In a sense, the browser searches for a style property spec, starting with inline, until it finds one (or there isn't one)
- Browsers can ignore style sheets or use their own default values for properties

3.2 Levels of Style Sheets (continued)

- Inline style sheets appear in the tag itself

- Document-level style sheets appear in the head of the document

- External style sheets are in separate files, potentially on any server on the Internet

  - Written as text files with the MIME type `text/css`

  - A `<link>` tag is used to specify that the browser is to fetch and use an external style sheet file

```
<link rel = stylesheet  type = "text/css"
 href = "http://www.wherever.org/termpaper.css">
</link>
```

- External style sheets can be validated

```
  http://jigsaw.w3.org/css-validator/
               validator-upload.html
```

3.3 Style Specification Formats

- Format depends on the level of the style sheet

- *Inline*:
  - Style sheet appears as the value of the `style` attribute
  - General form:

```
style = "property_1: value_1;
         property_2: value_2;
          …
         property_n: value_n"
```

  - Scope of an inline style sheet is the content of the tag

- *Document-level*:
  - Style sheet appears as a list of rules that are the content of a `<style>` tag

  - The `<style>` tag must include the `type` attribute, set to `"text/css"`

  - The list of rules must be placed in an HTML comment, because it is <u>not</u> HTML

  - Comments in the rule list must have a different form - use C comments (`/*…*/`)

3.3 Style Specification Formats (continued)

- General form:

```
<style type = "text/css">
 <!--
   rule list
 -->
</style>
```

- Form of the rules:
  selector {list of property/values}

  - The selector is a tag name or a list of tag names, separated by commas
    - Examples:

```
h1, h3
p
```

  - Each property/value pair has the form:
    property: value

  - Pairs are separated by semicolons, just as in the value of a `<style>` tag

- *External style sheets*

  - Form is a list of style rules, as in the content of a `<style>` tag for document-level style sheets

3.4 Style Classes

- Used to allow different occurrences of the same tag to use different style specifications

- A style class has a name, which is attached to a tag name

  - For example,

```
p.narrow {property/value list}
p.wide {property/value list}
```

- The class you want on a particular occurrence of a tag is specified with the `class` attribute of the tag

  - For example,

```
<p class = "narrow">
 ...
 </p>
 ...
 <p class = "wide">
 ...
 </p>
```

3.4 Style Classes (continued)

- A generic class can be defined if you want a style to apply to more than one kind of tag
  - A generic class must be named, and the name must begin with a period
    - Example,

```
.really-big { … }
```

- Use it as if it were a normal style class

```
<h1 class = "really-big"> … </h1>
 ...
<p class = "really-big"> … </p>
```

3.5 Properties & Property Values

- *There are 56 different properties in 6 categories*:
  - Fonts
  - Colors and backgrounds
  - Text
  - Boxes and layouts
  - Lists
  - Tags

- *Property Value Forms*

  - *Keywords* - `left`, `small`, …
    - Not case sensitive

  - *Length* - numbers, maybe with decimal points
    - Units:

      `px` - pixels
      `in` - inches
      `cm` - centimeters
      `mm` - millimeters
      `pt` - points
      `pc` - picas (12 points)
      `em` - height of the letter ‘m’
      `x-height` - height of the letter ‘x’

      - No space is allowed between the number and the unit specification e.g., `1.5 in` is illegal!

3.5 Properties & Property Values
(continued)

- Percentage - just a number followed immediately by a percent sign

- URL values
  - `url(`protocol`://`server`/`pathname`)`

- Colors
  - Color name
  - `rgb(`n1`,` n2`,` n3`)`
    - Numbers can be decimal or percentages
  - Hex form: `#`XXXXXX

- Property values are inherited by all nested tags, unless overriden

- *Font Properties*

  - `font-family`
    - Value is a list of font names - browser uses the first in the list it has
    - `font-family: Arial, Helvetica, Courier`
    - Generic fonts: `serif`, `sans-serif`, `cursive`, `fantasy`, and `monospace` (defined in CSS)
      - Browser has a specific font for each

3.5 Properties & Property Values
(continued)

- If a font name has more than one word, it should be single-quoted

- `font-size`
  - Possible values: a length number or a name, such as `smaller`, `xx-large`, etc.

- `font-style`
  - `italic`, `oblique` (useless), `normal`

- `font-weight` - degrees of boldness
  - `bolder`, `lighter`, `bold`, `normal`
    - Could specify as a multiple of 100 (100 – 900)

- `font`
  - For specifying a list of font properties

    `font: bolder 14pt Arial Helvetica`

  - Order must be: style, weight, size, name(s)

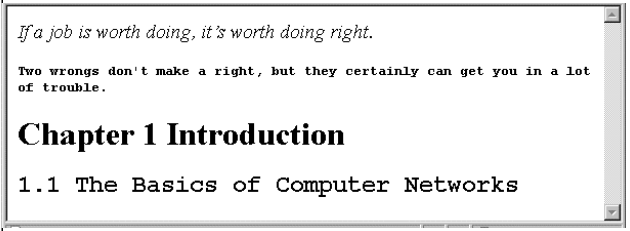
3.5 Properties & Property Values

(continued) - Example `fonts.html`)

```
<html>
<head>
   <title> Font Properties </TITLE>
   <style type = "text/css">
      <!-- /* Document-level style sheet*/
        p.big {font-size: 14pt;
               font-style: italic;
               font-family: 'Times New Roman';
              }
        p.small {font-size: 10pt;
                 font-weight: bold;
                 font-family: 'Courier New';
                }
       -->
   </style>
</head>
<body>
<p class = "big">
If a job is worth doing, it’s worth doing right.
</p>
<p class = "small">
Two wrongs don't make a right, but they certainly
can get you in a lot of trouble.
</p>
<h2 style = "font-family: 'Times New Roman';
             font-size: 24pt; font-weight: bold">
   Chapter 1 Introduction
</h2>
<h3  style = "font-family: 'Courier New';
              font-size: 18pt">
   1.1 The Basics of Computer Networks
</h3>
</body>
</html>
```

3.5 Properties & Property Values

(continued)



If a job is worth doing, it’s worth doing right.

Two wrongs don't make a right, but they certainly can get you in a lot of trouble.

Chapter 1 Introduction

1.1 The Basics of Computer Networks

- The `text-decoration` property
  - `line-through, overline, underline, none`

- *List properties* - `list-style-type`

  - *Unordered lists*
    - Bullet can be a disc (default), a square, or a circle

    - Set it on either the `<ul>` or `<li>` tag
      - On `<ul>`, it applies to list items

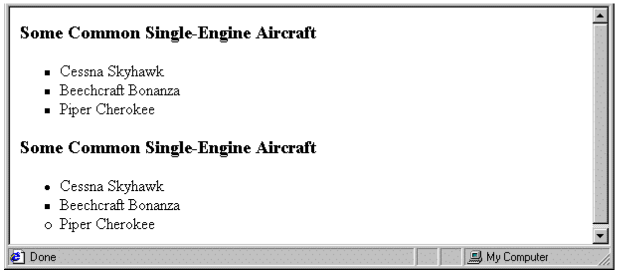
```
<h3> Some Common Single-Engine Aircraft </h3>
<ul style = "list-style-type: square">
    <li> Cessna Skyhawk </li>
    <li> Beechcraft Bonanza </li>
    <li> Piper Cherokee </li>
</ul>
```

3.5 Properties & Property Values

(continued)

- On `<li>`, `list-style-type` applies to just that item

```
<h3> Some Common Single-Engine Aircraft </h3>
<ul>
    <li style = "list-style-type: disc">
         Cessna Skyhawk </li>
    <li style = "list-style-type: square">
         Beechcraft Bonanza </li>
    <li style = "list-style-type: circle">
         Piper Cherokee </li>
</ul>
```



Some Common Single-Engine Aircraft

▪ Cessna Skyhawk
▪ Beechcraft Bonanza
▪ Piper Cherokee

Some Common Single-Engine Aircraft

• Cessna Skyhawk
▪ Beechcraft Bonanza
○ Piper Cherokee

3.5 Properties & Property Values

(continued)

- Could use an image for the bullets in an unordered list

  - Example:

```
<li style = "list-style-image:
             url(bird.jpg)">
```

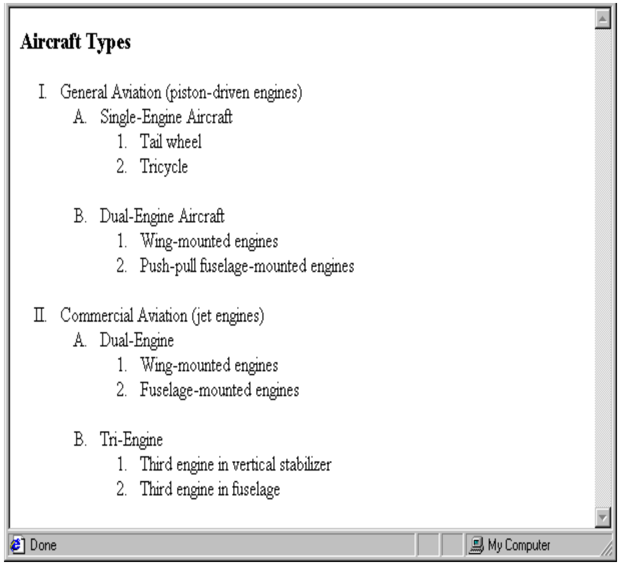
- *On ordered lists* - `list-style-type` can be used to change the sequence values

Property value	*Sequence type*	*First four*
`decimal`	Arabic numerals	1, 2, 3, 4
`upper-alpha`	Uc letters	A, B, C, D
`lower-alpha`	Lc letters	a, b, c, d
`upper-roman`	Uc Roman	I, II, III, IV
`lower-roman`	Lc Roman	i, ii, iii, iv

3.5 Properties & Property Values
(continued)

```
<h3> Aircraft Types </h3>
<ol style = "list-style-type: upper-roman">
    <li> General Aviation (piston-driven engines) </li>
    <ol style = "list-style-type: upper-alpha">
        <li> Single-Engine Aircraft </li>
        <ol style = "list-style-type: decimal">
            <li> Tail wheel </li>
            <li> Tricycle </li>
        </ol> <br />
     <li> Dual-Engine Aircraft </li>
     <ol style = "list-style-type: decimal">
         <li> Wing-mounted engines </li>
         <li> Push-pull fuselage-mounted engines </li>
     </ol>
    </ol> <br />
    <li> Commercial Aviation (jet engines) </li>
    <ol style = "list-style-type: upper-alpha">
        <li> Dual-Engine </li>
        <ol style = "list-style-type: decimal">
            <li> Wing-mounted engines </li>
            <li> Fuselage-mounted engines </li>
        </ol> <br />
        <li> Tri-Engine </li>
        <ol style = "list-style-type: decimal">
            <li> Third engine in vertical stabilizer
            </li>
             <li> Third engine in fuselage </li>
        </ol>
    </ol>
  </ol>
```

3.5 Properties & Property Values
(continued)



Aircraft Types

I. General Aviation (piston-driven engines)
   A. Single-Engine Aircraft
      1. Tail wheel
      2. Tricycle

   B. Dual-Engine Aircraft
      1. Wing-mounted engines
      2. Push-pull fuselage-mounted engines

II. Commercial Aviation (jet engines)
   A. Dual-Engine
      1. Wing-mounted engines
      2. Fuselage-mounted engines

   B. Tri-Engine
      1. Third engine in vertical stabilizer
      2. Third engine in fuselage

3.5 Properties & Property Values
(continued)

- *Alignment of text*

  - The `text-indent` property allows indentation

    - Takes either a length or a % value

  - The `text-align` property has the possible values, `left` (the default), `center`, `right`, or `justify`

  - Sometimes we want text to flow around another element - the `float` property

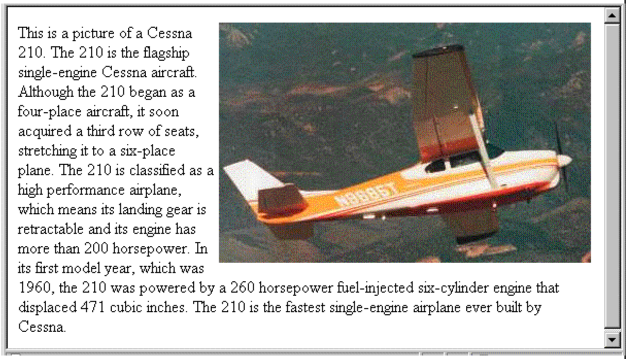
    - The `float` property has the possible values, `left`, `right`, and `none` (the default)

    - If we have an element we want on the right, with text flowing on its left, we use the default `text-align` value (`left`) for the text and the `right` value for `float` on the element we want on the right

3.5 Properties & Property Values
(continued)

```
<img src = "c210.jpg"
     style = "float: right" />
```

-- Some text with the default alignment - `left`



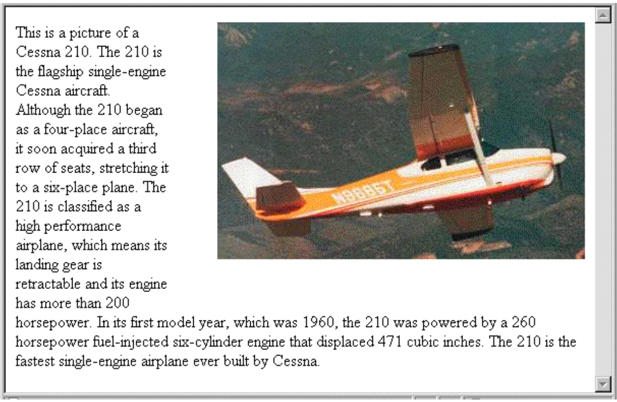
This is a picture of a Cessna 210. The 210 is the flagship single-engine Cessna aircraft. Although the 210 began as a four-place aircraft, it soon acquired a third row of seats, stretching it to a six-place plane. The 210 is classified as a high performance airplane, which means its landing gear is retractable and its engine has more than 200 horsepower. In its first model year, which was 1960, the 210 was powered by a 260 horsepower fuel-injected six-cylinder engine that displaced 471 cubic inches. The 210 is the fastest single-engine airplane ever built by Cessna.

3.5 Properties & Property Values
(continued)

- *Margins*

  - The margins around an HTML object can be set with `margin-left`, etc. - just assign them a length value

```
<img src = "c210.jpg " style = "float: right;
   margin-left: 0.5in; margin-bottom: 0.5in" />
```



This is a picture of a Cessna 210. The 210 is the flagship single-engine Cessna aircraft. Although the 210 began as a four-place aircraft, it soon acquired a third row of seats, stretching it to a six-place plane. The 210 is classified as a high performance airplane, which means its landing gear is retractable and its engine has more than 200 horsepower. In its first model year, which was 1960, the 210 was powered by a 260 horsepower fuel-injected six-cylinder engine that displaced 471 cubic inches. The 210 is the fastest single-engine airplane ever built by Cessna.

3.6 Colors

- *Color is a problem for the Web for two reasons*:

  1. Monitors vary widely
  2. Browsers vary widely

- There are three color collections

  1. There is a set of 16 colors that are guaranteed to be displayable by all graphical browsers on all color monitors

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

  2. There is a much larger set, the Web Palette

  - 216 colors

  - Use hex color values of `00`, `33`, `66`, `99`, `CC`, and `FF`

  - Inside back cover of this book has them!

3.6 Colors (continued)

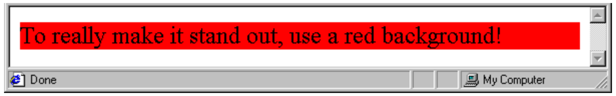
3. Any one of 16 million different colors

- The `color` property specifies the foreground color of elements

```
<table border = "5px">
  <tr>
    <th style = "color: red"> Apple </th>
    <th style = "color: orange"> Orange </th>
    <th style = "color: orange"> Screwdriver
     </th>
    </tr>
</table>
```

- The `background-color` property specifies the background color of elements

```
 <p style = "font-size: 24; color: blue;
             background-color: red">
 To really make it stand out, use a red
  background!
 </p>
```



To really make it stand out, use a red background!

- The `background-image` property

  - Use with care

3.7 The `<span>` and `<div>` tags

- One problem with the font properties is that they apply to whole elements, which are often too large

  - Solution: a new tag to define an element in the content of a larger element - `<span>`

  - The default meaning of `<span>` is to leave the content as it is

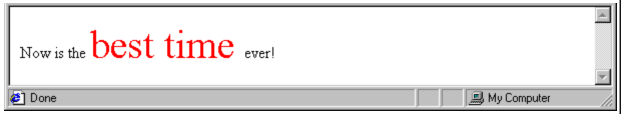
    - e.g.,

```
<p>
Now is the <span> best time </span> ever!
</p>
```

- Use `<span>` to apply an inline style sheet to its content

```
<p>
   Now is the
       <span style = "font-size: 40;
                      font-family: Ariel;
                      color: red">
   best time </span> ever!
</p>
```

3.7 The `<span>` and `<div>` tags (continued)



Now is the best time ever!

- The `<span>` tag is similar to other HTML tags, they can be nested and they have `id` and `class` attributes

- Another tag that is useful for style specifications: `<div>`
  - Used to create document sections (or divisions) for which style can be specified
    - e.g., A section of five paragraphs for which you want some particular style