Cascading Style Sheets

HTML is primarily concerned with content, rather than style.

CSS provide Web authors with a powerful and flexible way to control the **presentation details** of documents.

Learning Expectations:

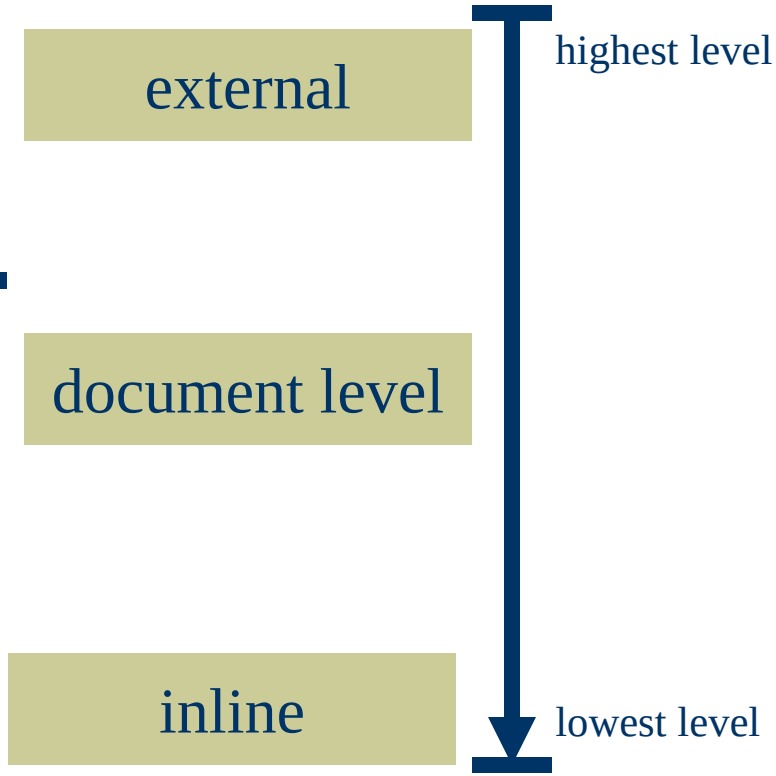
- What is CSS and is it useful?
- Three Levels of Style Sheets
- Style Specification Formats
- Style Classes
- Properties & Property Values
- CSS Inheritance

What is CSS?

- A styling language to impose consistency on the style of Web pages.
  - E.g., allow authors to specify that all occurrences of a particular tag use the same presentation style.
  - Not a new idea – word processors and desktop publishing systems have used style sheets for a long time.
- Website style sheets are called ***cascading*** style sheets because they can be defined at three different levels to specify the style of a document.
  - Lower-level style rules can override higher-level style rules, so the style of the content of a HTML element is determined through a cascade of styles.

3 Levels of Style Sheets

1. **Inline-level style sheets** apply to the content of a single HTML element.
2. **Document-level style sheets** apply to the whole body of a document.
3. **External-level style sheets** can apply to the bodies of any number of documents.



NOTE: The properties of a HTML element are those that result from a merge of all applicable style sheets, with lower-level style sheets having precedence in cases of conflicting instructions.

Style Specification Formats

- Inline styles should be used sparingly..
  - An inline style loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly, such as when a style is to be **applied to a single occurrence of an element**.
  - To use inline styles you use the style attribute in the relevant tag. The example shows how to change the colour and the left margin of a paragraph:

```
<p style="color:sienna; margin-left:20px;">
     This is a paragraph
</p>
```

Style Specification Formats

- **Document-Level**
  - Document style specifications appear as the content of a `<style>…</style>` element within the **head** of a document.

```
<!-- include list of rules below -->
<style>
/* List of rules go here */
</style>
```

NOTE: Comments within the style element start with /* and end with */.

```
<head><title> A title </title>
<style type="text/css">
          h1,h2,h3,h4,h5,h6{font-family:"sans serif";}
          h1{font-weight:bold; font-size:14pt;}
          h2{font-weight:lighter; font-size:12pt;}
          h1, h2{color:red}
</style>
</head>
```

Style Specification Formats

- **External Level**
  - With an external style sheet, you can change the look of an entire website by changing just one file!
  - External style sheets are not part of the documents to which they apply. They are stored separately and are specified by the documents that use them. The browser fetches external style sheets just as it fetches other Web pages. The **`link`** element is used to specify external style sheets. This appears in the head of the document.

```
<head>
<link rel="stylesheet" type="text/css"
      href="mystyle.css" />
</head>
```

Style Specification Formats

- **External Level**
  - In the previous example the browser will read the style definitions from the file **mystyle.css**, and format the document according to it.
  - An external style sheet can be written in any text editor. The file should not contain any html tags/elements. Your style sheet should be saved with a .css extension. An example of a style sheet file is shown below:

mystyle.css

```
hr{color:sienna;}
p{margin-left:20px;}
p.first-letter{color:blue;}
body{background-image:url("images/back40.gif");}
```

Style Specification Formats

- The syntax for rules in document and external level style sheets is the same.
  - Each rule is made up of three parts: a selector, a property and a value:

```
selector{property:value;}
```

  - The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value. The property and value are separated by a colon and surrounded by curly braces:

```
body{color:black;}
```

  - If the value is multiple words, put single-quotes around the value:

```
p{font-family:'sans serif';}
```

Style Specification Formats

- If you wish to specify more than one property, you should separate each property with a semi-colon. The example below shows how to define a center aligned paragraph, with a red text color:

```
tag_name{property1:v1; property2:v2; …; propertyN:vN;}
```

```
e.g., p{text-align:center; color:red;}
```

- To make the style definitions more readable, you can describe one property on each line, like this:

```
p
{
text-align:center;
color:black;
font-family:arial;
}
```

Style Specification Formats

- You can group selectors. Separate each selector with a comma. In the example below we have grouped all the header elements. Each header element will be green:

```
h1,h2,h3,h4,h5,h6 {color:green;}
```

- In this example the first line of each paragraph will appear in italics, as will the main (i.e. largest) heading.

```
p.first-line,h1 {font-style:italic; colour:blue;}
```

Style Classes

- With the class selector you can define different styles for the same type of HTML element. Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:

```
p.righty {text-align:right;}
p.centerish {text-align:center;}
```

- You have to use the class attribute in your HTML document:

```
<p class="righty">
   This paragraph will be right-aligned.
</p>

<p class="centerish">
   This paragraph will be center-aligned.
</p>
```

Style Classes

- Sometimes you may want a class of style specifications that apply to the content of more than one element. This can be done using a **generic class**, which is defined without an element name in its selector part. Instead you replace this with the name of the generic class, which **must begin with a fullstop.**

```
.italicy {font_style:italic;}
```

Now, in the body of the document you could have these elements…

```
<h3 class = "italicy"> This heading will be italicatised!
</h3>

<p class = "italicy">This paragraph will be italicatised!
</p>
```

Properties and Property Values

- Numerous properties exist in different categories such as fonts, backgrounds, lists, text, borders, icons, color, etc.
- We will look at just a few of these here. All details of all properties and property values can be found at www.w3c.org
- In the following slides we will see examples for how to represent the following: **basic font properties, basic list properties, alignment of text, margins and colour**.

Property Value Forms

- Nominal values - `left, right, large, small`, … Not case sensitive

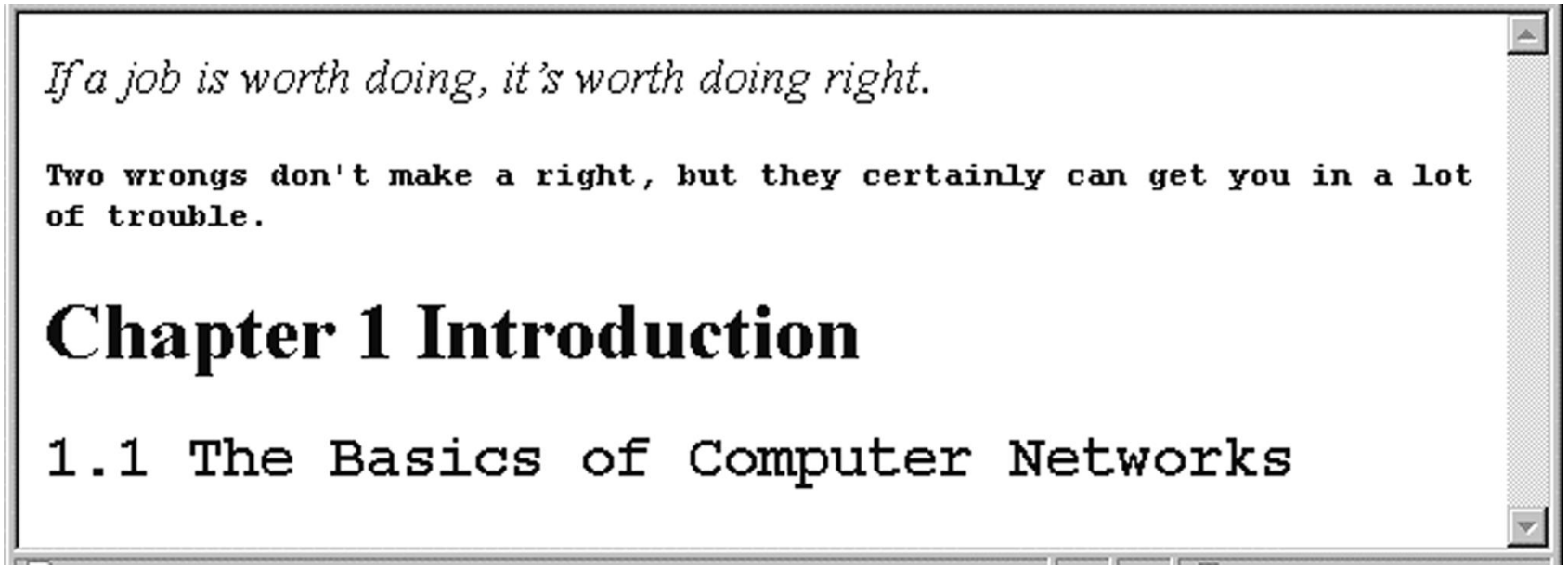
- Length Units – numeric values, may include decimal points
  - Some examples of allowable units include:

    px - pixels
    in - inches
    mm - millimeters
    cm - centimeters
    pt - points
    pc - picas (12 points)
    em - relative to the font-size of the element (2em means 2 times the size of the current font)
    ex (x-height) - relative to the height of the letter ‘x’ of current font

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

Font Properties

- Using style sheets for setting font properties.



Font Properties

- There are two types of font family names:
  - **generic family** – a collection of font families with similar characteristics
  - **font family** – a specific font family (e.g., Arial)

Generic Family	Font Family	Description
Serif	Times New Roman Georgia Constantia	Serif fonts have small lines on some characters.
Sansserif	Arial Helvetica Verdana	"Sans" means without – these fonts do not have the lines at the ends of the characters.
Monospaced	Courier Lucida Console	All monospace characters have the same width.

Font Properties

- The **font-family** property is used to specify a list of font names.

```
<h2 style = "font-family: 'Apple Chancery', Script, 'Times New
  Roman', Courier;"> This heading will be displayed in Apple
  Chancery if the browser supports this font, otherwise it will
  be displayed in Script, TNR, or Courier. </h2>
```

- The **font-style** property is most commonly used to specify italic.
  - You can specify values: `normal, italic, oblique`

```
<h2 style = "font-style: italic;" > Text is italicatised! </h2>
```

Font Properties

- The **font-weight** property is used to specify the degree of boldness.
  - You can specify values `normal`, `bold`, `bolder`, `lighter`.

```
<h2 style = "font-weight: bold;" > This will be bold. </h2>
```

- If more than one property must be specified, the values can be stated in a list as the value of the **font** property.

```
font: bold 14pt 'Times New Roman', Courier;
```

NOTE: The order in which the property values are given in a font value list is important. The **font style and weight must be firs**t, followed by font size, and finally the list of font names. **Only the font size and font family are required in the font list**.

Font Properties

- The **font-size** value can be absolute, or relative size.
  - Absolute size:
    - sets the text to specified size
    - does not allow a user to change the text size in all browsers
  - Relative size...
    - sets the size relative to surrounding elements
    - allows a user to change the text size in browsers

Examples of Allowable Units for **font-size** values

- “Ems”, Rems, Pixels, Points, Percent, Viewport units, etc.

Font Properties

- Examples of absolute units for **font-size** values
  - **Pixels (px):** fixed-size units where one pixel is equal to one dot on your computer screen.
  - **Points (pt):** traditionally used by print media, One point is equal to 1/72 of an inch. Fixed size, and does not scale in size.
- Key Point:
  - Fixed-sized units do not change based on the size of the viewport or are scalable.

Font Properties

- Some examples of relative units for **font-size** values

  - **“Ems” (em):** a scalable unit that is converted by the browser into pixel values. If the default font size in chrome is 16px, 1em = 16 pixels. Em is relative to the font size of the element on which they are used. Used for typography such as headings, text, paragraphs, margins, etc.

  - **Rem:** is always relative to the font size of the **html** element (aka. The ‘root’ element). If the font-size of the html element is 16 px, then 1 rem = 16px.

  - **Percent:** is always relative to another value. Most commonly used for controlling the width and height of containers.

Font Properties

- Points to note on **font-size** value units:
  - If you do not specify a font size, the default for normal text, like paragraphs, is 16px.
  - Generally 1em = 12pt = 16px = 100%
  - Both em and percent units get larger as the base font increases, but pixels and points do not.
  - It can be easy to set an absolute size for your text, but it's much easier on your visitors to use scalable text that can display on any device.

	`body { font-size: 100%; }`	`body { font-size: 120%; }`
`font-size: 1em`	The quick brown fox	The quick brown
`font-size: 12pt`	The quick brown fox	The quick brown fox
`font-size: 16px`	The quick brown fox	The quick brown fox
`font-size: 100%`	The quick brown fox	The quick brown

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    font-size: 100%;
}

h1 {
    font-size: 2.5em;
}

h2 {
    font-size: 1.875em;
}

p {
    font-size: 0.95em;
}
</style>
</head>
<body>

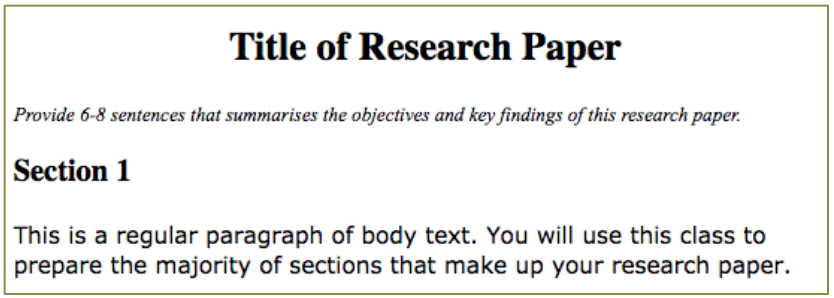
<h1>This is the largest heading </h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size using a combination of percent
and em units displays the same size in all major browsers,
and allows all browsers to resize the text!</p>

</body>
</html>
```

- An Example…
  - You can use a combination of units.
  - Set default size in percent for <body> element.
  - Result: shows the same text size in all browsers, and allows all browsers to zoom or resize text!

Font Properties

- Another Example…




```
<!DOCTYPE html>
<html>
<head>
<style>
p.abstract {
    font: italic 12px 'times new roman';
}

p.bodytext {
    font: normal 14px verdana, arial;
}
</style>
</head>
<body>

<h2 style="text-align:center;"> Title of Research Paper </h2>

<p class="abstract">Provide 6-8 sentences that summarises
the objectives and key findings of this research paper.</p>

<h3> Section 1 </h3>

<p class="bodytext">This is a regular paragraph of body
text. You will use this class to prepare the majority of
sections that make up your research paper. </p>

</body>
</html>
```

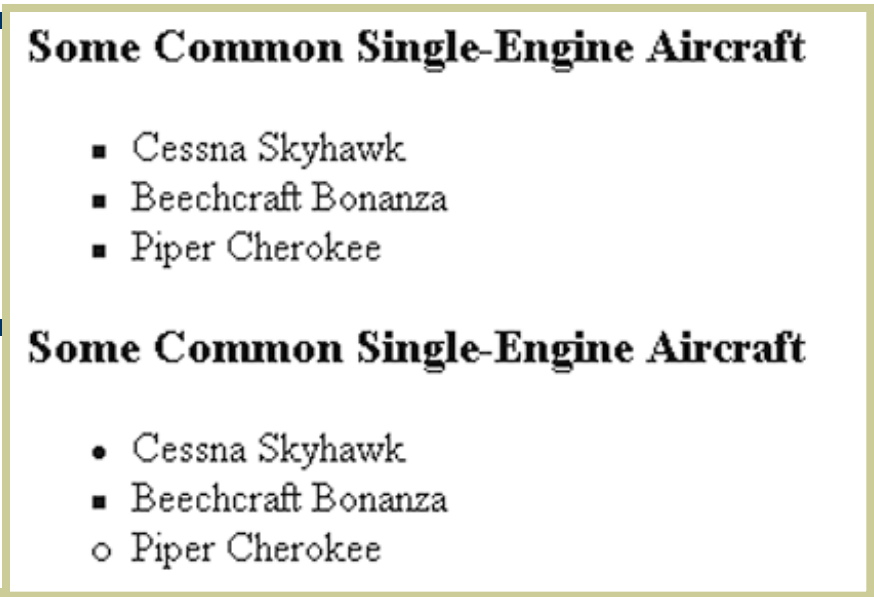
List Properties

- Two presentation details of lists often specified in HTML documents are:
  - The shape of bullets that precede the items in an unordered list, and
  - The sequencing values that precede the items in ordered lists.

- The **`list-style-type`** property is used for both of these.

List Properties

- Examples of unordered lists
  - Value can be a `disc` (default), a `square`, a `circle, etc.`




```
<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>
```

```
<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>
```

List Properties

- An example of nested ordered lists with different sequence values.

Aircraft Types

I. General Aviation (piston-driven engines)
  A. Single-Engine Aircraft
    1. Tail wheel
    2. Tricycle
II. Dual-Engine Aircraft
  1. Wing-mounted engines
  2. Push-pull fuselage-mounted engines
III. 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

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**

Possible sequencing values for ordered lists.

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

Special Elements

- …
  - Provides a way to include an inline style sheet that applies to a range of text that is smaller than a line or a paragraph.


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

- …
  - Provides a way to define sections of a document that can have their own style properties.

e.g. A section of x number of paragraphs for which you want some particular style.

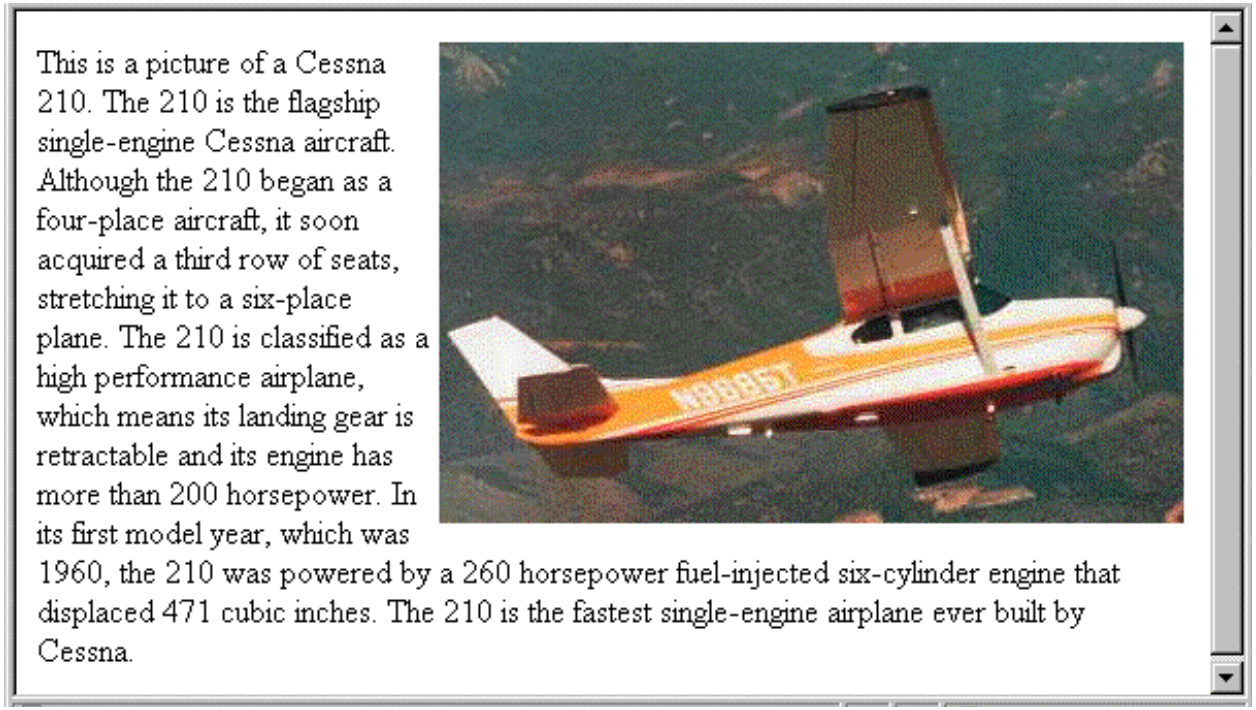
Alignment of Text

- 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.

Alignment of Text

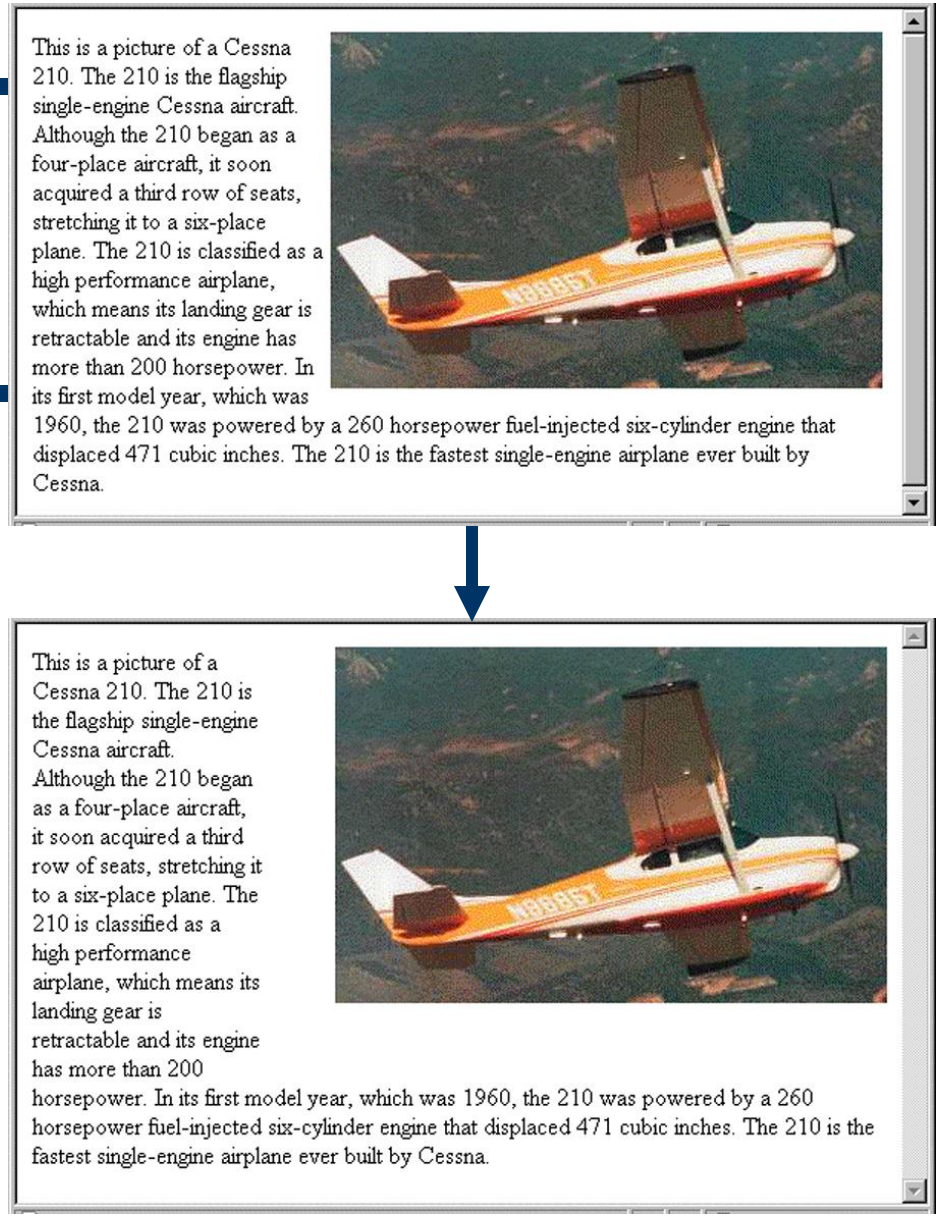
- This example features some text with the default left alignment.

```
<img src= "c21.jpg" alt="" style= "float: right;" />
```



Margins

- The margins around a HTML element can be set with the **`margin-left, margin-right, margin-bottom`** and **`margin-top`** properties.
- These need to be assigned a length value.




```
<img src = "c21.jpg " alt="" style="float:right;
                                    margin-left:2rem;
                                    margin-bottom:2rem;" />
```

Margins

- The **margin** property is a shorthand for the individual margin properties **`margin-top`**, **`margin-right`**, **`margin-bottom`** and **`margin-left`**.

```
p { margin: 100px 150px 100px 80px;}
```

- What if the **margin** property has less than four values?
  - Three values: **margin: 100px 50px 80px;**
  - Two values: **margin: 50px 25px;**
  - One value: **margin: 60px;**

Margins

- You can set the margin property to auto to horizontally center the element within its container.

- The element will take up the specified width, and the remaining space will be split equally between the left and right margins:

```
div {width:300px; margin:auto; border:5px solid green;}
```

Margins

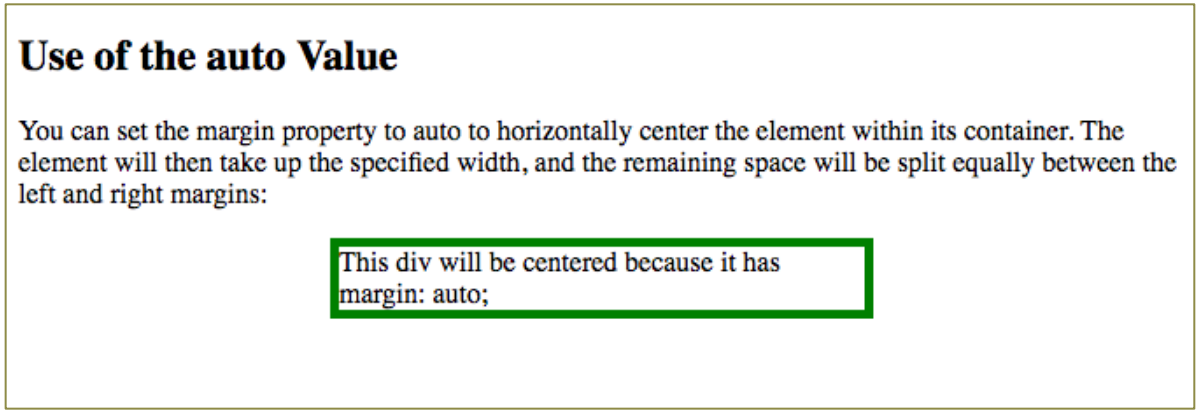
- A worked example of using the **`auto`** value…

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width:300px;
    margin: auto;
    border: 5px solid green;
}
</style>
</head>
<body>

<h2>Use of the auto Value</h2>
<p>You can set the margin property to auto to horizontally center the element
within its container.
The element will then take up the specified width, and the remaining space will
be split equally between the left and right margins:</p>

<div>
This div will be centered because it has margin: auto;
</div>

</body>
</html>
```



Colour

- Colors in CSS are most often specified by:
  - a valid color name (note: color names are case insensitive)
  - an RGB value
  - a HEX value in the form: (note: color names are also case insensitive)

Color Name	Red, Green, Blue (RGB) Model	Hexadecimal
red	rgb(255,0,0)	#FF0000
green	rgb(0,255,0)	#00FF00
blue	rgb(0,0,255)	#0000FF
orange	rgb(255,165,0)	#FFA500
yellow	rgb(255,255,0)	#FFFF00

- Currently, CSS supports 140 standard color names!

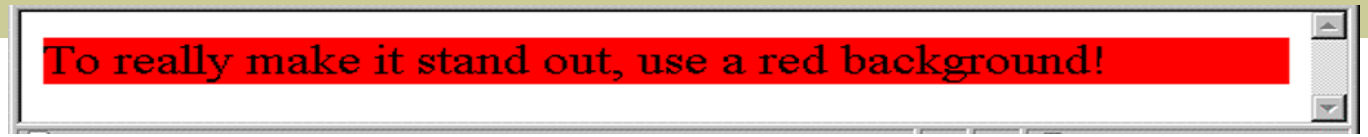
Colour

- The **`color`** property specifies the foreground colour of elements.

```
<table>
  <tr>
    <th style = "color:red;"> Apple </th>
    <th style = "color:orange;"> Orange </th>
    <th style = "color:#FFFF00;"> Screwdriver
    </th>
  </tr>
</table>
```

- The **`background-color`** property specifies the background colour of elements.

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



Inheritance in CSS

- If some properties have been set for the same selector in different style sheet levels, the values will be inherited from the more specific level.
  - For example, an external style sheet has these properties for the h3 selector:

```
h3{color:red; text-align:left; font-size:8pt;}
```

  - And an internal style sheet has these properties for the h3 selector:

```
h3{text-align:right; font-size:20pt;}
```

  - If the page with the internal style sheet also links to the external style sheet the properties for h3 will be:

```
color:red; text-align:right; font-size:20pt
```

- The colour is **inherited** from the external style sheet and the text-alignment and the font-size is replaced by the internal style sheet.

What's new in CSS3?

- CSS3 is the latest iteration of Cascading Style Sheet specification
- It is backwards compatible with CSS2; no need to update existing sites
- Some of the more important CSS3 modules are:
  - Selectors
  - Backgrounds and Borders
  - Text Effects
  - 2D/3D Transformations
  - Multiple Column Layout
  - User Interface

CSS3: Selectors

- Selectors are the patterns used to select the element you want to style

h1 {color:blue; font-size:12px;}

- Some new selectors in CSS3:
  - **element1~element2** e.g. p~ul Selects every <ul> element that are preceded by a <p> element
  - **[attribute^=value]** e.g., a[src^="https"] Selects every <a> element whose src attribute value begins with "https”
  - **[attribute$=value]** e.g., a[src$=".pdf"] Selects every <a> element whose src attribute value ends with ".pdf”
  - **[attribute*=value]** e.g., a[src*="COMP20030"] Selects every <a> element whose src attribute value contains the substring "COMP20030”
  - **:not(selector)** e.g., :not(p) Selects every element that is not a <p> element

See a full list of selectors available on http://www.w3schools.com/cssref/css_selectors.asp

Border Properties

- Previous versions of CSS have supported some border properties such as:

Property	Examples of Values	# of vals
border-style	dotted, dashed, solid, double, none, etc.	1-4
border-width	thin, medium, thick, or any size (in, cm, mm, px,pt, em, etc)	1-4
border-colour	red, green, blue, etc, or any RGB/Hex value	1-4

- Note the following ordering for multiple values:
  - 4 values: *top* border, *right* border, *bottom* border, *left* border
  - 3 values: : *top*, *right & left*, *bottom*
  - 2 values: *top & bottom, right & left*
  - 1 value: apply the same value to all borders

Border Properties

- Other important points to note:
  - The **border-style** property must be set before any of the others can ever be used.
  - The **border** property allows you to specify values for width, style and colour.

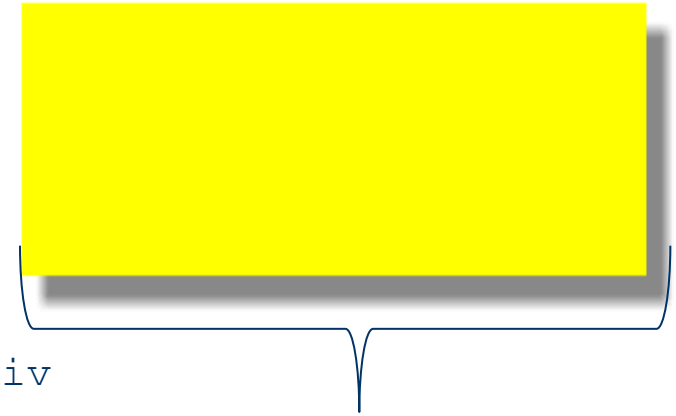
```
p { border: 5px solid red;}
```

CSS3: Borders

```
div
{
border:2px solid;
border-radius:25px;
}
```

- Adding rounded corners in CSS2 involved using different images for each corner.
- This is achieved in CSS3 using the **border-radius** property.

This box has rounded corners!



```
div
{
box-shadow: 10px 10px 5px #888888;
}
```

Value	Description
h-shadow	Required. The position of the horizontal shadow. Negative values are allowed
v-shadow	Required. The position of the vertical shadow. Negative values are allowed
blur	Optional. The blur distance
spread	Optional. The size of shadow
none	Deafult value. No shadow is displayed
color	Optional. The color of the shadow. The default value is black. Look at CSS Color Values for a complete list of possible color values. **Note:** In Safari (on PC) the color parameter is required. If you do not specify the color, the shadow is not displayed at all.
inset	Optional. Changes the shadow from an outer shadow (outset) to an inner shadow
initial	Sets this property to its default value. Read about *initial*
inherit	Inherits this property from its parent element. Read about *inherit*

CSS3: Borders

- If you only specify one value for the **border-radius property** the same radius will be applied to all four corners.
  - Four values example – applies clockwise assignment of radius values
  - Three values example - applies ‘Z’ assignment of radius values
  - Two values example - applies ‘X’ assignment of radius values

Property	Description
border-radius	A shorthand property for setting all the four border-*-*-radius properties
border-top-left-radius	Defines the shape of the border of the top-left corner
border-top-right-radius	Defines the shape of the border of the top-right corner
border-bottom-right-radius	Defines the shape of the border of the bottom-right corner
border-bottom-left-radius	Defines the shape of the border of the bottom-left corner

border-radius: 15px 50px 30px 5px;

border-radius: 15px 50px 30px;

border-radius: 15px 50px;

CSS3: Backgrounds

- Before CSS3, the background image size was determined by the actual size of the image.
- ```
  div
  {
  background:url(bg.gif);
  background-size:80px 60px;
  background-repeat:no-repeat;
  }
  ```


- ```
  div
  {
  background:url(bg.gif);
  background-repeat:no-repeat;
  background-size:100% 100%;
  background-origin:content-box;
  }
  ```
- CSS3 also allows you to use multiple background images for an element:

```
body
{
background:url(img_tree.gif),url(img_flwr.gif);
background-size:100% 100%;
background-repeat:no-repeat;
}
```

CSS3: Text Effects

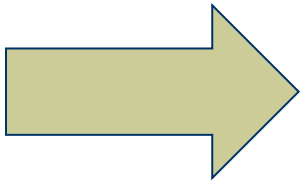
- ```
  h1
  {
  text-shadow: 5px 5px 5px #FF0000;
  }
  ```

Text shadow effect!

Value	Description
h-shadow	Required. The position of the horizontal shadow. Negative values are allowed
v-shadow	Required. The position of the vertical shadow. Negative values are allowed
blur	Optional. The blur distance
color	Optional. The color of the shadow. Look at CSS Color Values for a complete list of possible color values

- ```
  p {word-wrap:break-word;}
  ```

This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.



This paragraph contains a very long word: thisisaveryveryveryveryveryverylongword. The long word will break and wrap to the next line.

Without

With

CSS3: Fonts

- Before CSS3, web designers had to use fonts that were already installed on the user's computer.
- With CSS3, web designers can use whatever font they prefer.
- **@font-face** rule you must first define a name for the font (e.g. myFirstFont), and then point to the font file.

```
<!DOCTYPE html>
<html>
<head>
<title>CSS3 properties</title>
<style>
@font-face
{
font-family: myFirstFont;
src: url(sansation_light.woff);
}

div
{
font-family:myFirstFont;
}
</style>
</head>
<body>

<div>
With CSS3, websites can finally use
fonts other than the pre-selected
"web-safe" fonts.
</div>

<p><b>Note:</b> Internet Explorer 8
and earlier, do not support the
@font-face rule.</p>

</body>
</html>
```

CSS3: 2D Transformation

- ```
  div
  {
  transform: rotate(30deg);
  }
  ```


- ```
  div
  {
  transform: scale(2,4);
  }
  ```


- ```
  div
  {
  transform: skew(30deg,20deg);
  }
  ```


  ```

CSS3: 3D Transformation

- ```
  div
  {
  transform: rotateX(120deg);
  }
  ```


- ```
  div
  {
  transform: rotateY(130deg);
  }
  ```