



Cascading Style Sheets (CSS)

```
171 #content .article img.left.border {  
172     padding: 0 9px 9px 0;  
173     border-right: 1px dotted #999;  
174     border-bottom: 1px dotted #999; }  
175 #content .article blockquote {  
176     margin-left: 10px;  
177     padding-left: 10px;  
178     border-left: 3px solid #252525; }  
179 #content .article ul {  
180     padding-left: 1em;  
181     list-style-type: circle; }
```

Table of Contents

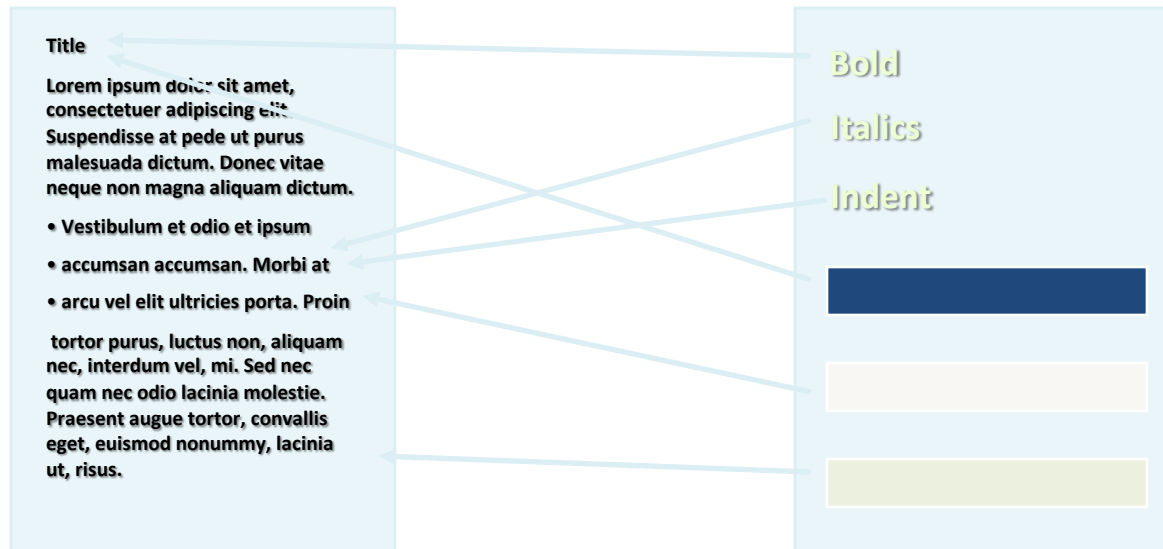
- What is CSS?
- Styling with Cascading Stylesheets (CSS)
- Selectors and style definitions
- Linking HTML and CSS
- Fonts, Backgrounds, Borders
- The Box Model
- Alignment, Z-Index, Margin, Padding
- Positioning and Floating Elements
- Visibility, Display, Overflow
- CSS Development Tools

CSS: A New Philosophy

- Separate content from presentation!

Content
(HTML document)

Presentation
(CSS Document)



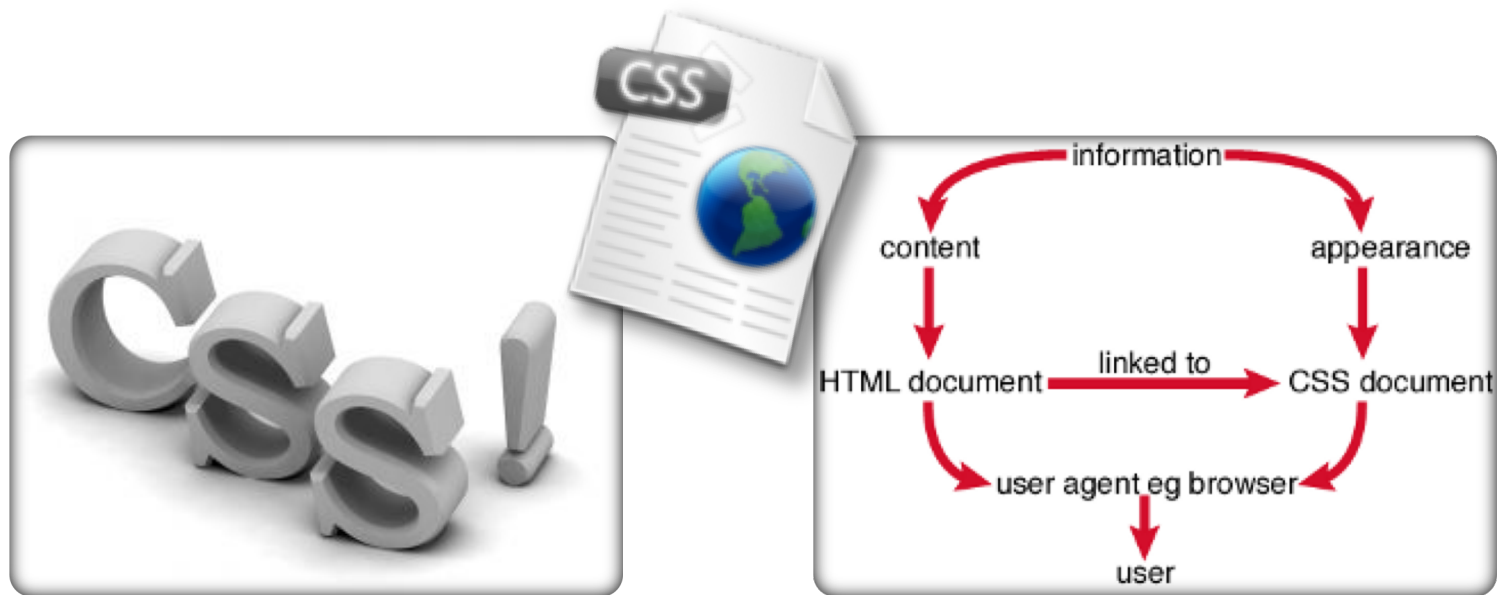
The Resulting Page

Title

**Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Suspendisse at pede ut purus
malesuada dictum. Donec vitae neque
non magna aliquam dictum.**

- ***Vestibulum et odio et ipsum***
- ***accumsan accumsan. Morbi at***
- ***arcu vel elit ultricies porta. Proin***

**Tortor purus, luctus non, aliquam nec,
interdum vel, mi. Sed nec quam nec
odio lacinia molestie. Praesent augue
tortor, convallis eget, euismod
nonummy, lacinia ut, risus.**



CSS Intro

Styling with Cascading Stylesheets

CSS Introduction

- Cascading Style Sheets (CSS)
 - Used to describe the presentation of documents
 - Define sizes, spacing, fonts, colors, layout, etc.
 - Improve content accessibility
 - Improve flexibility
- Designed to separate presentation from content
- Due to CSS, all HTML presentation tags and attributes are deprecated, e.g. font, center, etc.

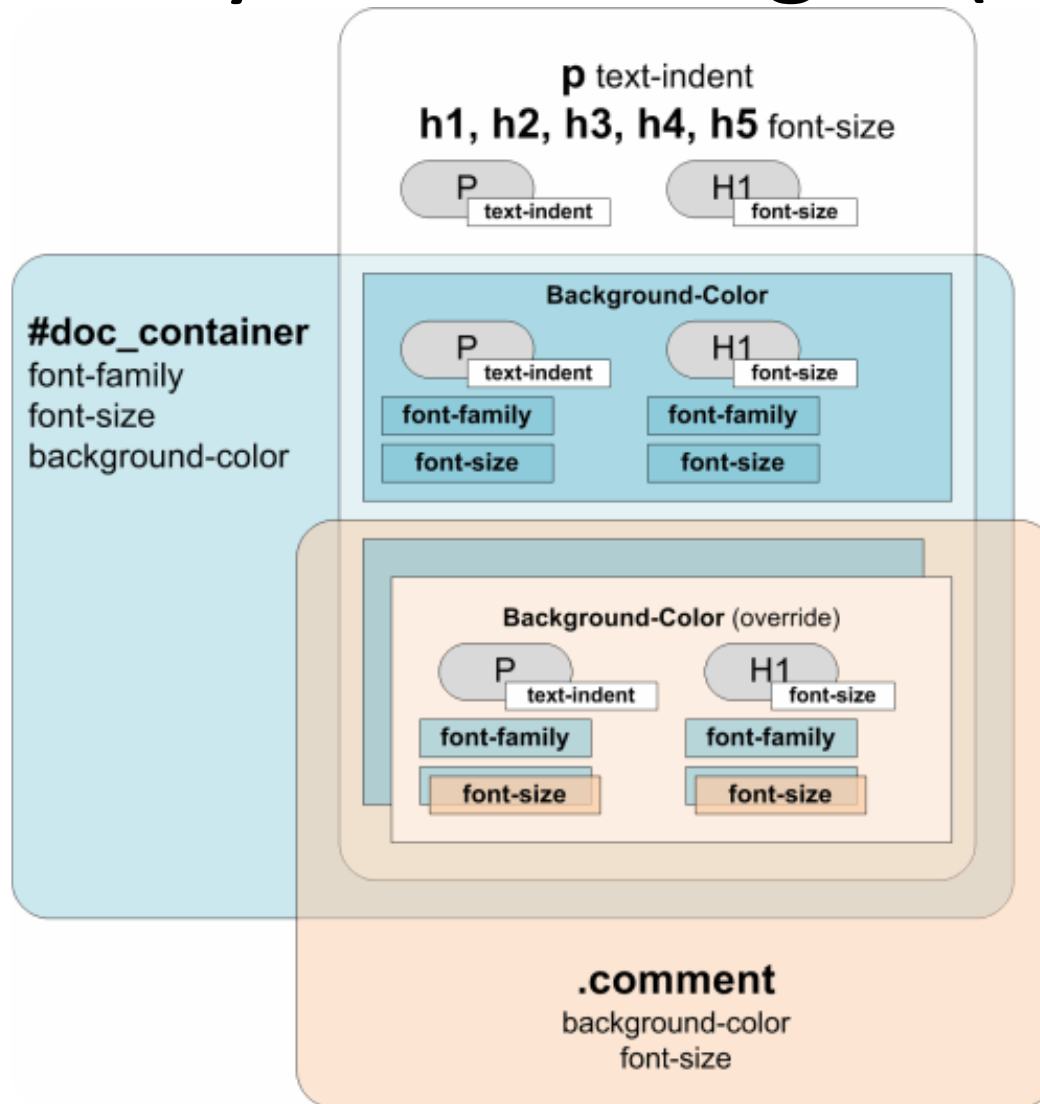
CSS Introduction (2)

- CSS can be applied to any XML document
 - Not just to HTML / XHTML
- CSS can specify different styles for different media
 - On-screen
 - In print
 - Handheld, projection, etc.
 - ... even by voice or Braille-based reader

Why “Cascading”?

- Priority scheme determining which style rules apply to element
 - **Cascade priorities** or **specificity (weight)** are calculated and assigned to the rules
 - Child elements in the HTML DOM tree inherit styles from their parent
 - Can override them
 - Control via `!important` rule

Why “Cascading”? (2)

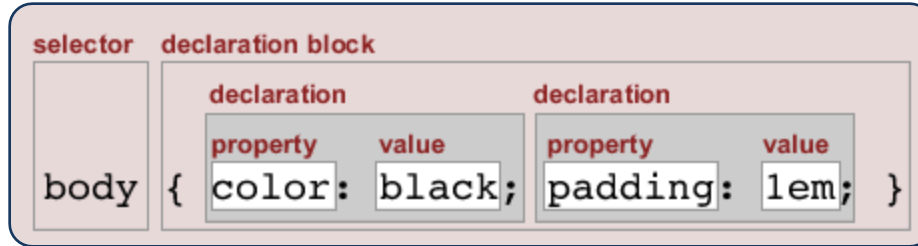


Why “Cascading”? (3)

- Some CSS styles are inherited and some not
 - Text-related and list-related properties are inherited - color, font-size, font-family, line-height, text-align, list-style, etc
 - Box-related and positioning styles are not inherited - width, height, border, margin, padding, position, float, etc
 - `<a>` elements do not inherit color and text-decoration

Style Sheets Syntax

- Stylesheets consist of rules, selectors, declarations, properties and values



<http://css.maxdesign.com.au/>

- Selectors are separated by commas
- Declarations are separated by semicolons
- Properties and values are separated by colons

h1,h2,h3 { color: green; font-weight: bold; }

Selectors

- Selectors determine which element the rule applies to:
 - All elements of specific type (tag)
 - Those that match a specific attribute (id, class)
 - Elements may be matched depending on how they are nested in the document tree (HTML)
- Examples:

```
.header a { color: green }
```

```
#menu>li { padding-top: 8px }
```

Selectors (2)

- Three primary kinds of selectors:
 - By tag (type selector):

```
h1 { font-family: verdana,sans-serif; }
```

- By element id:

```
#element_id { color: #ff0000; }
```

- By element class name (only for HTML):

```
.myClass {border: 1px solid red}
```

- Selectors can be combined with commas:

```
h1, .link, #top-link {font-weight: bold}
```

This will match `<h1>` tags, elements with class `link`, and element with id `top-link`

Selectors (3)

- Pseudo-classes define state
 - :hover, :visited, :active
- Pseudo-elements define element "parts" or are used to generate content
 - :first-line, :before, :after

```
a:hover { color: red; }  
p:first-line { text-transform: uppercase; }  
.title:before { content: "»"; }  
.title:after { content: "«"; }
```

Selectors (4)

- Match relative to element placement:

```
p a {text-decoration: underline}
```

This will match all <a> tags that are inside of <p>

- * – universal selector (avoid or use with care!):

```
p * {color: black}
```

This will match all descendants of <p> element

- + selector – used to match “next sibling”:

```
img + .link {float:right}
```

This will match all siblings with class name link that appear immediately after tag

Selectors (5)

- > selector – matches direct child nodes:

```
p > .error {font-size: 8px}
```

This will match all elements with class error, direct children of <p> tag

- [] – matches tag attributes by regular expression:

```
img[alt~=logo] {border: none}
```

This will match all tags with alt attribute containing the word logo

- .class1.class2 (no space) - matches elements with both (all) classes applied at the same time

Values in the CSS Rules

- Colors are set in RGB format (decimal or hex):
 - Example: `#a0a6aa` = `rgb(160, 166, 170)`
 - Predefined color aliases exist: `black`, `blue`, etc.
- Numeric values are specified in:
 - Pixels, ems, e.g. `12px` , `1.4em`
 - Points, inches, centimeters, millimeters
 - E.g. `10pt` , `1in`, `1cm`, `1mm`
 - Percentages, e.g. `50%`
 - Percentage of what?...
 - Zero can be used with no unit: `border: 0;`

Default Browser Styles

- Browsers have default CSS styles
 - Used when there is no CSS information or any other style information in the document
- Caution: default styles differ in browsers
 - E.g. margins, paddings and font sizes differ most often and usually developers reset them

```
* { margin: 0; padding: 0; }
```

```
body, h1, p, ul, li { margin: 0; padding: 0; }
```

Linking HTML and CSS

- HTML (content) and CSS (presentation) can be linked in three ways:
 - Inline: the CSS rules in the `style` attribute
 - No selectors are needed
 - Embedded: in the `<head>` in a `<style>` tag
 - External: CSS rules in separate file (best)
 - Usually a file with `.css` extension
 - Linked via `<link rel="stylesheet" href=...>` tag or `@import` directive in embedded CSS block

Linking HTML and CSS (2)

- Using external files is highly recommended
 - Simplifies the HTML document
 - Improves page load speed as the CSS file is cached

Inline Styles: Example

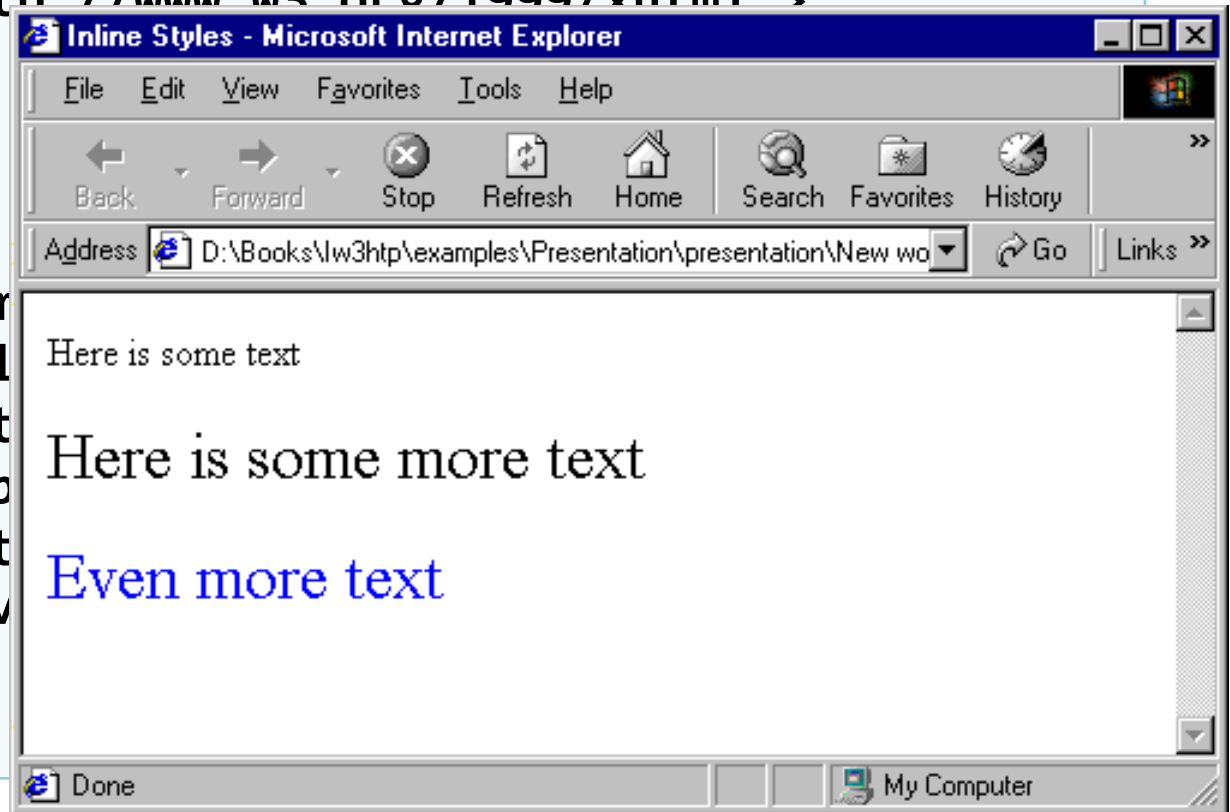
inline-styles.html

```
<!DOCTYPE>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Inline Styles</title>
</head>
<body>
  <p>Here is some text</p>
  <!--Separate multiple styles with a semicolon-->
  <p style="font-size: 20pt">Here is some
    more text</p>
  <p style="font-size: 20pt;color:
    #0000FF" >Even more text</p>
</body>
</html>
```

Inline Styles: Example

inline-styles.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Inline
</head>
<body>
  <p>Here is some text
  <!-- Separate multiple paragraphs -->
  <p style="font-size: 1.2em;">Here is some more text
  <p style="font-size: 1.2em; color: #0000FF;">Even more text
</body>
</html>
```



Embedded Styles

- Embedded in the HTML in the `<style>` tag:

```
<style type="text/css">
```

- The `<style>` tag is placed in the `<head>` section of the document
- `type` attribute specifies the MIME type
 - MIME describes the format of the content
 - Other MIME types include `text/html`, `image/gif`, `text/javascript` ...
- Used for document-specific styles

Embedded Styles: Example

embedded-stylesheets.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Style Sheets</title>
  <style type="text/css">
    em {background-color:#8000FF; color:white}
    h1 {font-family:Arial, sans-serif}
    p  {font-size:18pt}
    .blue {color:blue}
  </style>
</head>
```

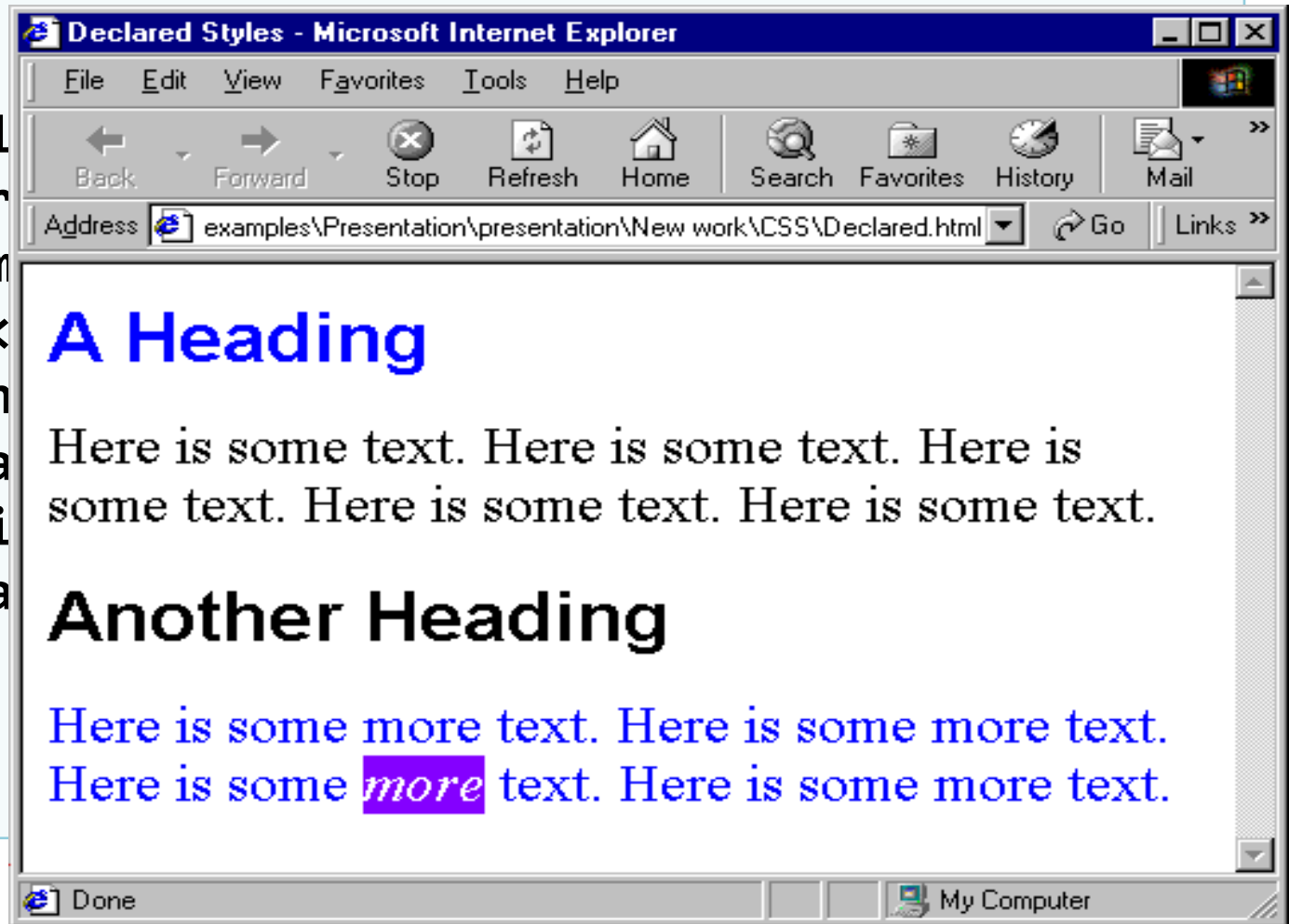

Embedded Styles: Example (2)

...

```
<body>
  <h1 class="blue">A Heading</h1>
  <p>Here is some text. Here is some text. Here
  is some text. Here is some text. Here is some
  text.</p>
  <h1>Another Heading</h1>
  <p class="blue">Here is some more text.
  Here is some more text.</p>
  <p class="blue">Here is some <em>more</em>
  text. Here is some more text.</p>
</body>
</html>
```

Embedded Styles: Example (3)

```
...  
<body>  
  <h1 class="c1">A Heading  
  <p>Here is some text.</p>  
  <h1>Another Heading  
  <p class="c2">Here is some text.</p>  
  <p class="c3">Here is some more text. Here is some more text.  
</body>  
</html>
```



External CSS Styles

- External linking
 - Separate pages can all use a shared style sheet
 - Only modify a single file to change the styles across your entire Web site
- `link` tag (with a `rel` attribute)
 - Specifies a relationship between current document and another document

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

- `link` elements should be in the `<head>`

External CSS Styles (2)

@import

- Another way to link external CSS files
- Example:

```
<style type="text/css">  
  @import url("styles.css");  
  /* same as */  
  @import "styles.css";  
</style>
```

- Ancient browsers do not recognize @import
- Use @import in an external CSS file to workaround the IE 32 CSS file limit

External Styles: Example

styles.css

```
/* CSS Document */

a          { text-decoration: none }

a:hover    { text-decoration: underline;
              color: red;
              background-color: #CCFFCC }

li em      { color: red;
              font-weight: bold }

ul          { margin-left: 2cm }

ul ul      { text-decoration: underline;
              margin-left: .5cm }
```

External Styles: Example (2)

external-styles.html

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Importing style sheets</title>
  <link type="text/css" rel="stylesheet"
        href="styles.css" />
</head>
<body>
  <h1>Shopping list for <em>Monday</em>:</h1>
  <li>Milk</li>
  ...

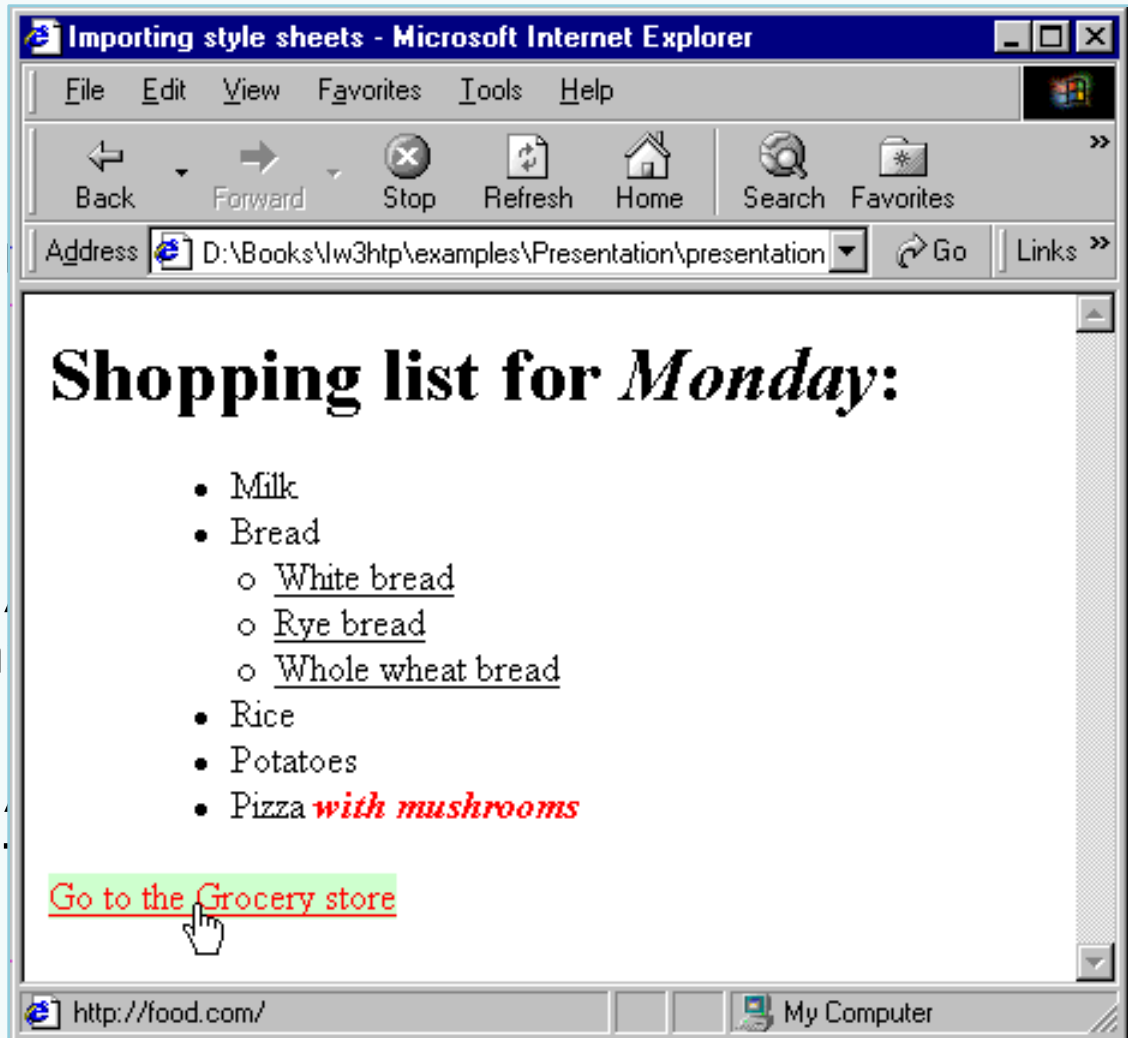
```

External Styles: Example (3)

```
...  
<li>Bread  
  <ul>  
    <li>White bread</li>  
    <li>Rye bread</li>  
    <li>Whole wheat bread</li>  
  </ul>  
</li>  
<li>Rice</li>  
<li>Potatoes</li>  
<li>Pizza <em>with mushrooms</em></li>  
</ul>  
<a href="http://food.com" title="grocery  
  store">Go to the Grocery store</a>  
</body>  
</html>
```

External Styles: Example (4)

```
...  
<li>Bread  
  <ul>  
    <li>White  
    <li>Rye b  
    <li>Whole  
  </ul>  
</li>  
<li>Rice</li>  
<li>Potatoes<  
<li>Pizza <em  
</ul>  
<a href="http://  
  store">Go to  
</body>  
</html>
```



Text-related CSS Properties

- `color` – specifies the color of the text
- `font-size` – size of font: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`, `smaller`, `larger` or numeric value
- `font-family` – comma separated font names
 - Example: `verdana`, `sans-serif`, etc.
 - The browser loads the first one that is available
 - There should always be at least one generic font
- `font-weight` can be `normal`, `bold`, `bolder`, `lighter` or a number in range `[100 ... 900]`

CSS Rules for Fonts (2)

- `font-style` – styles the font
 - Values: `normal`, `italic`, `oblique`
- `text-decoration` – decorates the text
 - Values: `none`, `underline`, `line-through`, `overline`, `blink`
- `text-align` – defines the alignment of text or other content
 - Values: `left`, `right`, `center`, `justify`

Shorthand Font Property

- **font**
 - Shorthand rule for setting multiple font properties at the same time

```
font:italic normal bold 12px/16px verdana
```

is equal to writing this:

```
font-style: italic;  
font-variant: normal;  
font-weight: bold;  
font-size: 12px;  
line-height: 16px;  
font-family: verdana;
```

Backgrounds

- `background-image`
 - URL of image to be used as background, e.g.:

```
background-image:url("back.gif");
```
- `background-color`
 - Using color and image and the same time
- `background-repeat`
 - `repeat-x`, `repeat-y`, `repeat`, `no-repeat`
- `background-attachment`
 - `fixed` / `scroll`

Backgrounds (2)

- `background-position`: specifies vertical and horizontal position of the background image
 - Vertical position: `top`, `center`, `bottom`
 - Horizontal position: `left`, `center`, `right`
 - Both can be specified in percentage or other numerical values
 - Examples:
 - `background-position: top left;`
 - `background-position: -5px 50%;`

Background Shorthand Property

- **background**: shorthand rule for setting background properties at the same time:

```
background: #FFF0C0 url("back.gif") no-repeat  
fixed top;
```

is equal to writing:

```
background-color: #FFF0C0;  
background-image: url("back.gif");  
background-repeat: no-repeat;  
background-attachment: fixed;  
background-position: top;
```

- Some browsers will not apply BOTH color and image for background if using shorthand rule

Background-image or ?

- Background images allow you to save many image tags from the HTML
 - Leads to less code
 - More content-oriented approach
- All images that are not part of the page content (and are used only for “beautification”) should be moved to the CSS

Borders

- `border-width`: `thin`, `medium`, `thick` or numerical value (e.g. `10px`)
- `border-color`: color alias or RGB value
- `border-style`: `none`, `hidden`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`
- Each property can be defined separately for left, top, bottom and right
 - `border-top-style`, `border-left-color`, ...

Border Shorthand Property

- `border`: shorthand rule for setting border properties at once:

```
border: 1px solid red
```

is equal to writing:

```
border-width: 1px;  
border-color: red;  
border-style: solid;
```

- Specify different borders for the sides via shorthand rules: `border-top`, `border-left`, `border-right`, `border-bottom`
- When to avoid `border: 0`

Width and Height

- `width` – defines numerical value for the width of element, e.g. `200px`
- `height` – defines numerical value for the height of element, e.g. `100px`
 - By default the height of an element is defined by its content
 - Inline elements do not apply height, unless you change their `display` style.

Margin and Padding

- `margin` and `padding` define the spacing around the element
 - Numerical value, e.g. `10px` or `-5px`
 - Can be defined for each of the four sides separately - `margin-top`, `padding-left`, ...
 - `margin` is the spacing outside of the border
 - `padding` is the spacing between the border and the content
 - What are collapsing margins?

Margin and Padding: Short Rules

- `margin: 5px;`
 - Sets all four sides to have margin of 5 px;
- `margin: 10px 20px;`
 - top and bottom to 10px, left and right to 20px;
- `margin: 5px 3px 8px;`
 - top 5px, left/right 3px, bottom 8px
- `margin: 1px 3px 5px 7px;`
 - top, right, bottom, left (clockwise from top)
- Same for padding

CSS-Box Model

