

## Cascading Style Sheets Benefits:

- Let you *assign several properties at once* to all similar text elements on you page. Controls color, font, size, background, etc.
- *Saves time*. One note can be used for all instances in a page. Or one small file can be created that sets the style for 100s of pages!



## Cascading Style Sheets Benefits:

- Easy to *change*:
  - One note for a whole page
  - One page for a whole web site
- Assures *consistency*
- Many *types of typographic control* besides bold & italic: font family, line spacing (leading), background color for a line of text, etc.
- Other *types of control* including location in window, rollover effects and movement



## CSS “Local”

- Affects only one piece of text
- The only CSS found in the <body>
- `<p style="background:yellow; color:red; border:double medium green"> Here's my text! </p>`



## CSS Internal or“Embedded”

- Affects all similar items in the whole page
- `<head>`
  - `<style> h1 {text-align:center; letter-spacing:.5em; background:green; color:yellow}`
  - `</style>`
- `</head>`



## CSS Internal or “Embedded”

- `<head>`  
    `<style>` `p {text-align:justify;`  
        `text-indent:8pt;`  
        `font:10pt/15pt “Helvetica”}`  
    `</style>`  
    `</head>`



## CSS External

```
h1 {text-align:center;
    letter-spacing:.5em;
    background:green;
    color:yellow}
```

- Use no other HTML tags. No `<style>` tags needed.
- Simple text file. Save as `name.css` (example: “`jb1.css`”)
- Each page that uses this external style must have a `<link>` tag in the `<head>`...
  - `<head>`  
    `<link rel= “stylesheet” type= “text/css”`  
        `href= “jb1.css” />`  
    `</head>`



## Which Will Get Precedence?

- Inline style *overrides* embedded style.
- Embedded style *overrides* external style.
- The “cascading” part:
  - Any style details you do not define locally will still be controlled by the style above it.
  - So you could change the font size of a particular `<p>` element, for instance, and the font family and indentation and color would still be controlled by the external stylesheet.



## CSS Classes (Option 1)

- Define more than just one h1 or one p, etc. Create separate versions of each.
- `p.subhead { font-family: Arial, sans-serif;`  
    `font-weight:bold; color:#FF0000 }`  
    `p.caption { font-family: Arial, sans-serif;`  
        `color:#000000 }`
- In the `<body>`, rather than a simple `<p>` use `<p class=“subhead”>My Text</p>`
- Classes allow you to plan a comprehensive stylesheet.



## CSS Classes (Option 2)

- Create a whole new class not based on an existing tag.

```
.question { font-family:"Times New Roman", Times, serif; font-size:12pt; font-weight:bold }
```

- In the <body>, use  
`<div class="question">My Text?</div>`



## Style Elements:

- font-family:
  - Times Arial serif sans-serif
- font-style:
  - Italic, normal, or oblique
- font-weight:
  - bold light extra-bold extra-light medium



## More Style Elements:

- font-size:24pt
  - small medium large, pt size, percentage
  - 12pt/15pt
    - (12 point with 15 pts of leading)
- line-height:1.2
  - by points or percentage or multiplying factor
- color:#0000FF
  - by colorname, hex, rgb or rgb%



## More Style Elements:

- background:blue
  - (color name or hex value)
- text-indent:2em
  - (8pt 5% 1em)
- text-align:center
  - (left right center justify)
- margin-left:12pt
  - (12pt 10% 3em)
  - Also margin-top -bottom -right



## Options

```
h1 { font-family: Arial, Helvetica,  
      sans-serif }
```

- *In this case, if Arial is not found on the user's system, and then if Helvetica is not found, another sans-serif font will be substituted.*



## Options

- Relative Measurements
  - 1.5em means 1.5x font size.
  - 10% as in margin:10%
- *Using relative measurements is smart. Then, even if the user has pre-set styles, your font measurements will be relatively correct and look appropriate. This makes the best use of the cascading characteristics.*



## DHTML (Dynamic HTML)

Client-side scripts modify elements of a page: Java Script + HTML

- Allows you to re-define tags
- Allows you to hide/display elements in response to local events like mouse-overs
- Allows for positioning options for elements like blocks of text or images, even stacking and motion
- Offers more style options

