

Cascading Style Sheets (CSS)

Web Engineering Class

Department of Computer Engineering, Junior of first semester 2009

본 내용은 University of Washington, CSE 190 M (Web Programming), Spring 2007 수업에서 사용된 자료입니다.

Except where otherwise noted, the contents of this presentation are © Copyright 2007 Marty Stepp and are licensed under the Creative Commons Attribution 2.5 License.



The bad way to produce styles

```
<p><font face="Arial">Welcome to Greasy Joe's.  You will
<b>never, <i>ever, <u>EVER</u></i></b> beat <font
size="+1" color="red">OUR</font> prices!</font></p>
```

Welcome to Greasy Joe's. You will **never**, *ever*, EVER beat **OUR** prices!

- the above tags such as `b`, `i`, `u`, and `font` are legal in older HTML but are deprecated in strict XHTML
 - you should not use the above tags on your homework assignments!
- why are we discouraged from expressing stylistic information this way?

Cascading Style Sheets (CSS)

- describe the appearance, layout, and presentation of information on a web page (as opposed to HTML, which describes the content of the page)
- describe *how* information is to be displayed, not *what* is being displayed
- can be embedded in HTML document or placed into separate `.css` file
 - advantage of `.css` file: one style sheet can be shared across many HTML documents

Basic CSS rule syntax

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

```
    property: value;
}

p {
    font-family: sans-serif;
    color: red;
}
```

- a CSS file consists of one or more rules
- each rule starts with a selector that specifies an HTML element and then applies style properties to it

Attaching a CSS file: <link>

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

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

```
<link rel="stylesheet" type="text/css"
href="http://www.google.com/uds/css/gsearch.css" />
```

- the link tag appears in the head of an HTML page
- can link to multiple style sheet files
 - in case of a conflict (two sheets define a style for the same HTML element), the latter sheet's properties will be used

CSS properties for colors

```
p {
    color: red;
    background-color: yellow;
}
```

This paragraph uses the style above.

- color: color of the element's text
- background-color: color that will appear behind the element

Specifying colors

```
p { color: red; }
h2 { color: rgb(128, 0, 196); }
h4 { color: #FF8800; }
```

This paragraph uses the first style above.

This heading uses the second style above.

This heading uses the third style above.

-
- color names: `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white` (`white`), `yellow`
 - RGB codes: red, green, and blue values from 0 (none) to 255 (full)
 - hex codes: RGB values in base-16 from 00 (0, none) to FF (255, full)

CSS properties for fonts

- font-family: which font will be used
- font-size: how large the letters will be drawn
- font-style: used to enable/disable italic style
- font-weight: used to enable/disable bold style
- Complete list of font properties

font-family

```
p { font-family: "Georgia"; }  
h2 { font-family: "Arial Narrow"; }
```

This paragraph uses the first style above.

This heading uses the second style above.

-
- enclose multi-word font names in quotes

More about **font-family**

```
p { font-family: "Garamond", "Times New Roman", serif; }
```

This paragraph uses the above style.

-
- can specify multiple fonts from highest to lowest priority
 - generic font names:
`serif`, `sans-serif`, `cursive`, `fantasy`, `monospace`
 - if the first font is not found on the user's computer, the next is tried
 - generally should specify similar fonts
 - placing a generic font name at the end of your `font-family` value ensures that every

computer will use a valid font

font-size

```
p { font-size: 14pt; }
```

This paragraph uses the style above.

More about font-size

```
p { font-size: x-large; }
```

This paragraph uses the above style.

- vague font sizes: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, **`xx-large`**
- relative font sizes: `smaller`, `larger`
- percentage font sizes, e.g.: `90%`, `120%`
- units: pixels (`px`) vs. point (`pt`) vs. m-size (`em`)
`16px`, `16pt`, `1.16em`
- `px` specifies a number of pixels on the screen (absolute)
- `pt` specifies number of *point*, where a point is 1/72 of an inch onscreen
- `em` specifies number of *m-widths*, where 1 `em` is equal to the font's current size

font-weight, font-style

```
p {  
  font-weight: bold;  
  font-style: italic;  
}
```

This paragraph uses the style above.

- either of the above can be set to `normal` to turn them off (e.g. headings)

Body styles

```
body { font-size: 16px; }
```

- to apply a style to the entire body of your page, write a selector for the `body` element
- saves you from manually applying a style to each element

Practice problem: Kittens



KITTENS!



Edit this HTML and add the following styles to it:

- entire page should have a pink background and use 16 point font
- main heading should use Comic Sans MS font
- lists should appear in a Lucida Console font
- list numbers should have yellow background; list items should have green background
- link text should be purple
- quote text should be italicized

Why I love them:

1. They are *little*.
2. They make **adorable** sounds:
 - "Meow!"
 - "Purr!"
3. JUST LOOK AT THEM!

Show HTML

Show Expected Appearance

Why ``, `` and not ``, `<i>`?

```
strong { font-weight: normal; color: red; }
em { font-style: normal; background-color: #DDDDDD; }
```

Now if I want to **strongly emphasize** something or just emphasize it, it doesn't necessarily have to be bold or italic.

- `strong` and `em` describe attributes of the content (it is something important in the document that you want to emphasize)
- `b` and `i` describe formatting and presentation ("I want this to be bold.")

CSS properties for text

- `text-align`: alignment of text within its element

- text-decoration: decorations such as underlining
- line-height, word-spacing, letter-spacing: gaps between the various portions of the text
- text-indent: indents the first letter of each paragraph
- Complete list of text properties

text-align

```
blockquote { text-align: justify; }  
h2 { text-align: center; }
```

The Emperor's Quote

[TO LUKE SKYWALKER] The alliance... will die. As will your friends. Good, I can feel your anger. I am unarmed. Take your weapon. Strike me down with all of your hatred and your journey towards the dark side will be complete.

-
- `text-align` can be `left`, `right`, `center`, or `justify` (which widens all full lines of the element so that they occupy its entire width)

text-decoration

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

This paragraph uses the style above.

-
- can also be overline, ~~line-through~~, `blink`
 - effects can be combined:

text-decoration: overline underline;

CSS properties for dimensions

```
p { width: 400px; background-color: yellow; }  
h2 { width: 50%; background-color: aqua; }
```

This paragraph uses the first style above.

This heading uses the
second style above.

- width, height: how wide or tall to make this element
- max-width, max-height, min-width, min-height: the maximum or minimum size of this element in the given dimension
- all of these apply only to block elements; ignored for inline elements

CSS comments: `/* ... */`

```
/* This is a comment.  
   It can span many lines in the CSS file. */  
p { color: red; background-color: aqua; }
```

- CSS (like HTML) is usually not commented as rigorously as programming languages such as Java
- the `//` single-line comment style is NOT supported in CSS

Grouping styles

```
p,h1,h2 { color: blue; }  
h2 { background-color: yellow; }
```

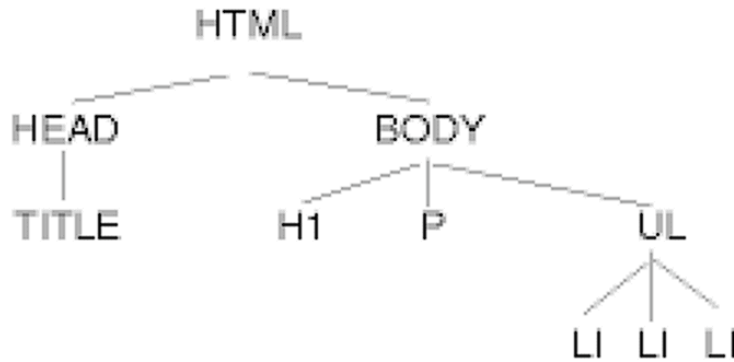
This paragraph uses the above style.

This heading uses the above style.

-
- a style can select multiple elements separated by commas
 - the given properties will be applied to all of the elements
 - the individual elements can also have their own styles (like `h2` above)

Document tree

```
<html><head><title>My home page</title></head>  
<body><h1>My home page</h1>  
<p>Let me tell you about my favorite composers:</p>  
<ul><li>Elvis Costello</li>  
<li>Johannes Brahms</li>  
<li>Georges Brassens</li>  
</ul></body></html>
```



Inheriting styles (explanation)

```

body { font-family: sans-serif; background-color: yellow; }
p { color: red; background-color: aqua; }
a { text-decoration: overline underline; }
h2 { font-weight: bold; text-align: center; }

```

This is a heading.

A styled paragraph. Previous slides are available on the web site.

- a bullet list
- when multiple styles apply to an element, they are inherited
- a more tightly matching rule can override a more general inherited rule
- not all properties are inherited (notice link's color above)

Styles that conflict

```

p,h1,h2 { color: blue; font-style: italic; }
h2 { color: red; background-color: yellow; }

```

This paragraph uses the first style above.

This heading uses both styles above.

- when two styles set conflicting values for the same property, the latter style takes precedence

W3C CSS Validator

```

<p><a href="http://jigsaw.w3.org/css-validator/check/referer">
<img src="http://jigsaw.w3.org/css-validator/images/vcss"

```


alt="Valid CSS!" /></p>



- jigsaw.w3.org/css-validator/
- checks your CSS to make sure it meets the official CSS specifications
- more picky than the web browser, which may render malformed CSS correctly

Practice problem: More kittens



KITTENS!



Edit the previously-styled Kitten HTML and add the following styles:

- all headings should be centered, bolded, and underlined
- the images should be enlarged to occupy one-third of the screen each
- the list of items should be narrowed to occupy only half the page width
- the text should be spaced so that the lines are further apart
- emphasized and strongly emphasized text should appear slightly larger than the other text on the page

Why I love them:

1. They are *little*.
2. They make **adorable** sounds:
 - "Meow!"
 - "Purr!"
3. JUST LOOK AT THEM!

Show HTML

Show Expected Appearance

CSS class selectors

```
p.special {
  background-color: yellow;
  font-weight: bold;
}
```

- selectively applies a CSS rule to only the paragraphs that are part of the class named `special` (next slide)
- gives a style to some occurrences of an element but not others ("I don't want ALL paragraphs to be yellow, just these three...")

The HTML `class` attribute

```
<p>Spatula City! Spatula City!</p>
<p class="special">See our spectacular spatula specials!</p>
<p class="special">Today only: satisfaction guaranteed.</p>
<p>We'll beat any advertised price!</p>
```

Spatula City! Spatula City!

See our spectacular spatula specials!

Today only: satisfaction guaranteed.

We'll beat any advertised price!

Class selectors without element

```
.standout {
  color: red;
  font-family: cursive;
}
```

- selectively applies a CSS rule to *any element* that is part of the class named `standout`

HTML `class` attribute revisited

```
<h2 class="standout">Spatula City! Spatula City!</h2>
<p class="special">See our spectacular spatula specials!</p>
<p class="special standout">Satisfaction guaranteed.</p>
<p class="standout">We'll beat any advertised price!</p>
```

Spatula City! Spatula City!

See our spectacular spatula specials!

Satisfaction guaranteed.

We'll beat any advertised price!

- an element can be a member of multiple classes (separated by spaces)

CSS ID selectors

```
p#missionstatement {
```

```
font-style: italic;
font-family: "Garamond", "Century Gothic", serif;
}
```

- selectively applies a CSS rule to only the paragraphs that has the particular ID named `missionstatement` (next slide)
- differs from class selector in that an ID can only be used once in the HTML document (won't validate otherwise)
- element (p above) can be omitted if desired; rule will apply to any element with ID `missionstatement`

The HTML `id` attribute

```
<p>Spatula City!  Spatula City!</p>
<p id="missionstatement">Our mission is to provide the most
spectacular spatulas and splurge on our specials until our
customers <q>esplode</q> with splendor!</p>
```

Spatula City! Spatula City!

Our mission is to provide the most spectacular spatulas and splurge on our specials until our customers "esplode" with splendor!

Linking to sections of a web page

```
<p>Visit <a href=
"http://www.textpad.com/download/index.html#downloads">
textpad.com</a> to get the TextPad editor.</p>
<p><a href="#mac">Directions for Mac OS X</a></p>
```

Visit [textpad.com](http://www.textpad.com) to get the TextPad editor.

[Directions for Mac OS X](#)

- a link target can include an ID at the end, preceded by a #
- browser will load that page and scroll to element with given ID
- can link to an ID within the current page

Logical divisions in HTML: `<div>`

a section or division of your HTML page (block-level)

```
<div class="standout">
<h2>Spatula City!  Spatula City!</h2>
<p class="special">See our spectacular spatula specials!</p>
<p>We'll beat any advertised price!</p>
```

</div>

Spatula City! Spatula City!

See our *spectacular spatula specials!*

We'll beat any advertised price!

- has no onscreen appearance, but you can apply a style or ID to it, which will be inherited by all elements inside the `div`

Inline styling sections:

an inline element used purely as a range for applying styles

```
<h2>Spatula City! Spatula City!</h2>
<p>See our <span class="special">spectacular</span>
spatula specials!</p>
<p>We'll beat <span class="standout">any advertised
price</span>!</p>
```

Spatula City! Spatula City!

See our **spectacular** spatula specials!

We'll beat *any advertised price!*

- has no onscreen appearance, but you can apply a style or ID to it, which will be applied to the text inside the `span`

Embedding style sheets: <style>

```
<head>
<style type="text/css">
p { font-family: sans-serif; color: red; }
h2 { background-color: yellow; }
</style>
</head>
```

- should be placed within the `head` of the HTML page
- linking to an external style sheet file is preferred, especially when you have many styles

Inline styles: the `style` attribute

```
<p style="font-family: sans-serif; color: red;">
This is a paragraph</p>
```

This is a paragraph

- higher precedence than embedded or linked styles
- useful for one-time overrides

Cascading style sheets

- it's called Cascading Style Sheets because the attributes of an element *cascade* together in this order:
 - browser's default styles
 - external style sheet files (in a `<link>` tag)
 - internal style sheets (inside a `<style>` tag in the page's header)
 - inline style (the `style` attribute of the HTML element)

CSS properties for backgrounds

- `background-color` : color to fill background
- `background-image` : image to place in background
- `background-position` : placement of bg image within element
- `background-repeat` : whether/how bg image should be repeated
- `background-attachment` : whether bg image scrolls with page
- `background` : shorthand to set all background properties

background-image

```
body {  
    background-image: url("draft.jpg");  
}
```

This is the first paragraph

This is the second paragraph...
It occupies 2 lines

-
- background image/color fills the content area and the padding

background-repeat

```
body {  
    background-image: url("draft.jpg");  
    background-repeat: repeat-x;  
}
```

This is the first paragraph

This is the second paragraph...
It occupies 2 lines

- can be repeat (default), repeat-x, repeat-y, or no-repeat

background-position

```
body {  
  background-image: url("draft.jpg");  
  background-repeat: no-repeat;  
  background-position: 370px 20px;  
}
```

This is the first paragraph

This is the second paragraph...
It occupies 2 lines

- value consists of two tokens, each of which can be top, left, right, bottom, center, a percentage, or a length value in px, pt, etc.
- value can be negative to shift left/up by a given amount

Showing a partial image

```
.partialimage1, .partialimage2 {  
  background-image: url("sex_and_the_city.jpg");  
  background-repeat: no-repeat;  
  width: 70px; height: 200px;  
}  
.partialimage1 { background-position: 0px 0px; }  
.partialimage2 { background-position: -115px 0px; }
```





Context selectors

```
selector1 selector2 {  
    properties  
}
```

- applies the given properties to selector2 only if it is inside a selector1 on the page

```
selector1 > selector2 {  
    properties  
}
```

- applies the given properties to selector2 only if it is *directly* inside a selector1 on the page (selector1 tag is immediately inside selector2 with no tags in between)

Context selector example

```
li strong { text-decoration: underline; }
```

```
<p>Shop at <strong>Hardwick's Hardware</strong>...</p>  
<ul>  
<li>The <strong>best</strong> prices in town!</li>  
<li>Act while supplies last!</li>  
</ul>
```

Shop at **Hardwick's Hardware...**

- The **best** prices in town!
- Act while supplies last!

More complex example

```
div#ad li.important strong { text-decoration: underline; }
```

```
<div id="ad">  
<p>Shop at <strong>Hardwick's Hardware</strong>...</p>  
<ul>  
<li class="important">The <strong>best</strong>  
prices in town!</li>  
<li>Act <strong>while supplies last!</strong></li>  
</ul>  
</div>
```

Shop at **Hardwick's Hardware...**

- The **best** prices in town!
- Act **while supplies last!**

Pseudo-classes

```
a:link {color: #FF0000}      /* unvisited link */
a:visited {color: #00FF00}   /* visited link */
a:hover {color: #FF00FF}     /* mouse over link */
a:active {color: #0000FF}    /* selected link */
```

- **:active** : an activated or selected element
- **:focus** : an element that has the keyboard focus
- **:hover** : an element that has the mouse over it
- **:link** : a link that has not been visited
- **:visited** : a link that has already been visited
- **:first-child** : an element that is the first child of another

Pseudo-class example

```
a:link {color: red}
a:visited {color: green}
a:hover {color: purple; background-color: yellow;}
a:active {color: blue}
```

```
<a href="http://www.google.com">Goooooogle</a>
```

Goooooogle

- How would we make unvisited links blue, only they are in a paragraph inside of the div with id header or id footer?
- answer:

```
div#header p a:link, div#footer p a:link {
    color: blue;
}
```

The list-style-type property

```
ol { list-style-type: lower-roman; }
```

- Possible values:
 - i. none : No marker
 - ii. disc (default), circle, square
 - iii. decimal : 1, 2, 3, etc.
 - iv. decimal-leading-zero : 01, 02, 03, etc.
 - v. lower-roman : i, ii, iii, iv, v, etc.
 - vi. upper-roman : I, II, III, IV, V, etc.
 - vii. lower-alpha : a, b, c, d, e, etc.
 - viii. upper-alpha : A, B, C, D, E, etc.

- ix. lower-greek : alpha, beta, gamma, etc.
- x. others: hebrew, armenian, georgian, cjk-ideographic, hiragana, katakana, hiragana-iroha, katakana-iroha

The display property

```
h2 { display: inline; background-color: yellow; }
```

This is a heading This is another heading

- sets the type of CSS box model an element is displayed with
- can be none, inline, block, run-in, compact, ...
- use sparingly, because it can radically alter the page layout

The visibility property

```
p.secret {  
    visibility: hidden;  
}
```

- sets whether an element should be shown onscreen
 - the element will still take up space onscreen, but will not be shown
 - to make it not take up any space, set display to none instead
- can be visible (default) or hidden
- can be used to show/hide dynamic HTML content on the page in response to events