Web Programming Step by Step

Chapter 3
CSS for Styling

3.1: Basic CSS

- **3.1: Basic CSS**
- 3.2: More CSS

The bad way to produce styles

```
<p>
  <font face="Arial">Welcome to Greasy Joe's.</font>
  You will <b>never</b>, <i>ever</i>, <u>EVER</u> beat
  <font size="+4" color="red">OUR</font> prices!
</p>
```

HTML

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

output

- tags such as `b`, `i`, `u`, and `font` are discouraged in strict XHTML
  - Why is this bad?

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
  - with a `.css` file, one style sheet can be shared across many pages

Basic CSS rule syntax (3.1.1)

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

CSS

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

CSS

- a CSS file consists of one or more **rules**
- each rule starts with a **selector** that specifies an HTML element(s) and then applies style **properties** to them
  - a selector of `*` selects all elements

Attaching a CSS file: `<link>` (3.1.2)

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

HTML

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

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

HTML

- a 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

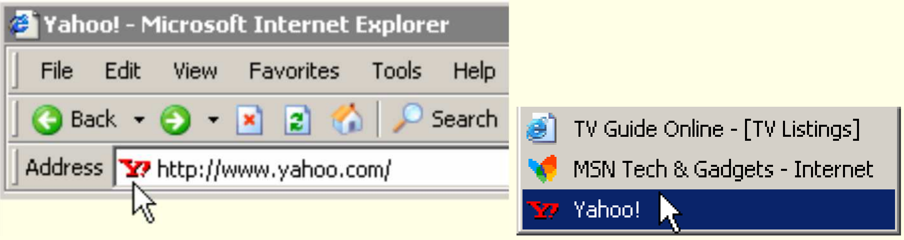
Aside: Favorites icon ("favicon")

```
<link href="filename" type="MIME type" rel="shortcut icon" />
```

HTML

```
<link href="yahoo.gif" type="image/gif" rel="shortcut icon" />
```

HTML


Yahoo! - Microsoft Internet Explorer
File
Edit
View
Favorites
Tools
Help
Back
Search
Address
http://www.yahoo.com/
TV Guide Online - [TV Listings]
MSN Tech & Gadgets - Internet
Yahoo!

output

- the `link` tag, placed in the HTML page's `head` section, can specify an icon for a web page
    - this icon will be placed in the browser title bar and bookmark/favorite
- IE6: this doesn't work; must place a file in `.ico` format named `favicon.ico` in the root directory of the web server (instructions)

Embedding style sheets: `<style>`

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

HTML

- CSS code can be embedded within the `head` of an HTML page
- this is *bad style* and should be avoided when possible (why?)

Inline styles: the `style` attribute

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

HTML

This is a paragraph

output

- higher precedence than embedded or linked styles
- used for one-time overrides and styling a particular element
- this is *bad style* and should be avoided when possible (why?)

CSS properties for colors (3.1.3)

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

CSS

This paragraph uses the style above.

output

property	description
`color`	color of the element's text
`background-color`	color that will appear behind the element

Specifying colors

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

CSS

This paragraph uses the first style above.

This h2 uses the second style above.

This h4 uses the third style above.

output

- color names: `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `lime`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, (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)

Grouping styles

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

CSS

This paragraph uses the above style.

This h2 uses the above styles.

output

- a style can select multiple elements separated by commas
- the individual elements can also have their own styles (like `h2` above)

CSS comments: /* ... */ (3.1.4)

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

- 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
- the `<!-- ... -->` HTML comment style is also NOT supported in CSS

CSS properties for fonts (3.1.5)

property	description
`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: "Courier New";
}
```

CSS

This paragraph uses the first style above.

This h2 uses the second style above.

output

- enclose multi-word font names in quotes

More about `font-family`

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

CSS

This paragraph uses the above style.

output

- 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;
}
```

CSS

This paragraph uses the style above.

output

- units: pixels (px) vs. point (pt) vs. m-size (em)
  16px, 16pt, 1.16em
- vague font sizes: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
- percentage font sizes, e.g.: 90%, 120%

- pt specifies number of *point*, where a point is 1/72 of an inch onscreen
- px specifies a number of pixels on the screen
- 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;
}
```

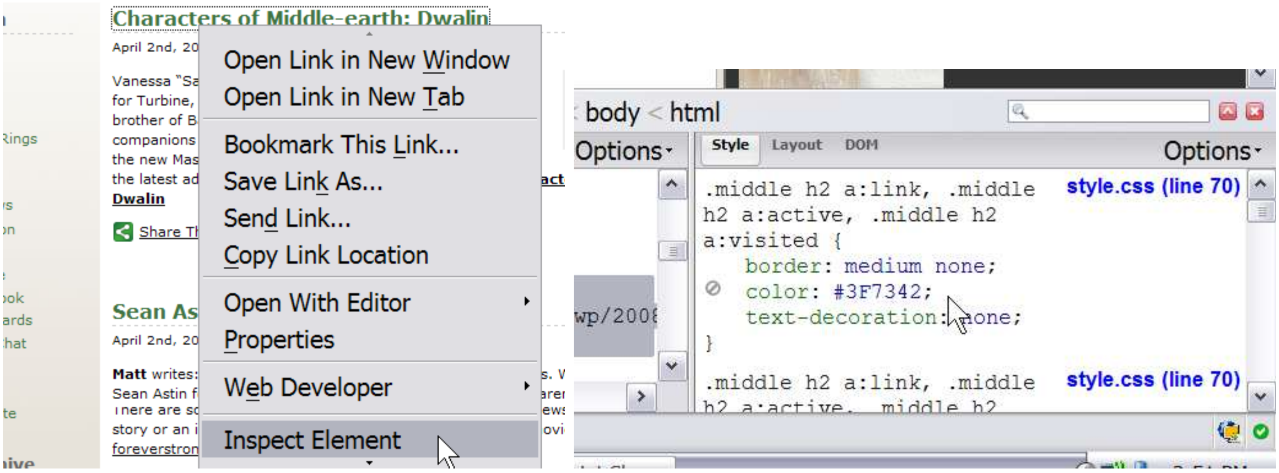
CSS

This paragraph uses the style above.

output

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

Spying on styles with Firebug



```
.middle h2 a:link, .middle
h2 a:active, .middle h2
a:visited {
    border: medium none;
    color: #3F7342;
    text-decoration: none;
}
.middle h2 a:link, .middle
h2 a:active, .middle h2
```

- install Firebug Firefox add-on to inspect XHTML/CSS of any page
- can see what styles apply to which elements, and change them dynamically

Why `<strong>`, `<em>` and not `<b>`, `<i>`?

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

CSS

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

output

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

CSS properties for text (3.1.6)

property	description
`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; }
```

CSS

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.

output

- `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;
}
```

CSS

This paragraph uses the style above.

output

- can also be overline, ~~line-through~~, blink, or none
- effects can be combined:

```
text-decoration: overline underline;
```

The list-style-type property (3.2.4)

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

CSS

- 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

3.2: More CSS

- 3.1: Basic CSS
- **3.2: More CSS**

Body styles

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

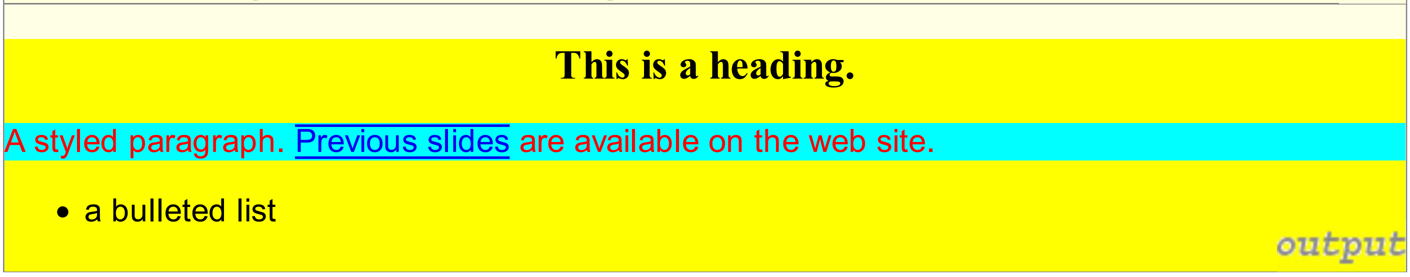
- to apply a style to the entire body of your page, write a selector for the `body` element
- saves you from manually applying a style to each element

Cascading style sheets

- it's called Cascading Style Sheets because the properties 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)

Inheriting styles (explanation) (3.2.1)

```
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 bulleted 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; }
```

CSS

This paragraph uses the first style above.

This heading uses both styles above.

output

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

The HTML `id` attribute (3.2.2)

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

HTML

Spatula City! Spatula City!

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

output

- allows you to give a unique ID to any element on a page
- each ID must be unique; can only be used once in the page

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="#mission">View our Mission Statement</a></p>
```

HTML

Visit textpad.com to get the TextPad editor.

View our Mission Statement

output

- 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

CSS ID selectors

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

CSS

Spatula City! Spatula City!

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

output

- applies style only to the paragraph that has the ID of `mission`
- element can be specified explicitly: `p#mission {`

The HTML `class` attribute (3.2.3)

```html
<p class="standout">Spatula City!  Spatula City!</p>
<p class="special">See our spectacular spatula specials!</p>
<p class="special">Today only: satisfaction guaranteed.</p>
```

HTML

Spatula City! Spatula City!

See our spectacular spatula specials!

Today only: satisfaction guaranteed.

output

- classes are a way to group some elements and give a style to only that group (“I don't want ALL paragraphs to be yellow, just these three...”)
- unlike an `id`, a `class` can be reused as much as you like on the page

CSS class selectors

```css
.special {
  background-color: yellow;
  font-weight: bold;
}
p.standout {
  color: red;
  font-family: cursive;
}
```

CSS

Spatula City! Spatula City!

See our spectacular spatula specials!

Today only: satisfaction guaranteed.

output

- applies rule to any element with class `special` or a `p` with class `standout`

Multiple classes

```html
<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>
```

HTML

Spatula City! Spatula City!

See our spectacular spatula specials!

Satisfaction guaranteed.

We'll beat any advertised price!

output

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

CSS pseudo-classes

```css
a:link    { color: #FF0000; }      /* unvisited link */
a:visited { color: #00FF00; }      /* visited link */
a:hover   { color: #FF00FF; }      /* mouse over link */
```

CSS

Buy early, buy often!

output

class	description
`: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-letter`	the first letter of text inside an element
`:first-line`	the first line of text inside an element
`:first-child`	an element that is the first one to appear inside another

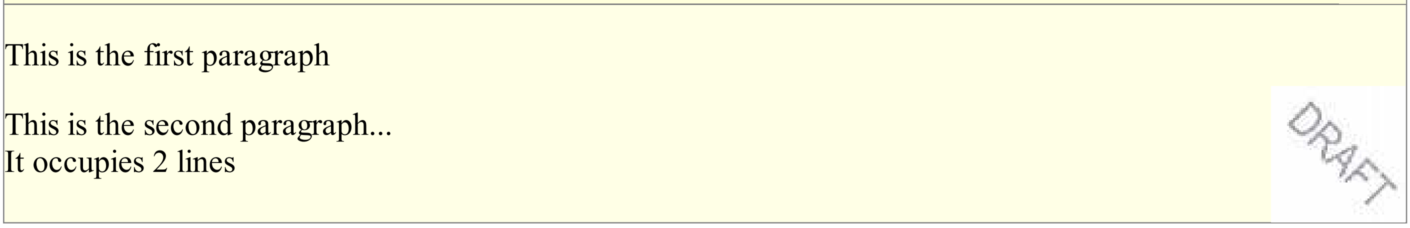
CSS properties for backgrounds

property	description
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

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

CSS



This is the first paragraph

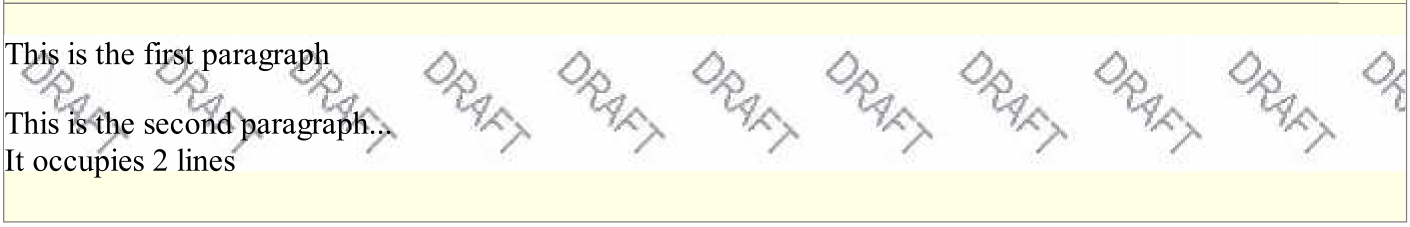
This is the second paragraph...
It occupies 2 lines

- background image/color fills the element's content area

background-repeat

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

CSS



This is the first paragraph

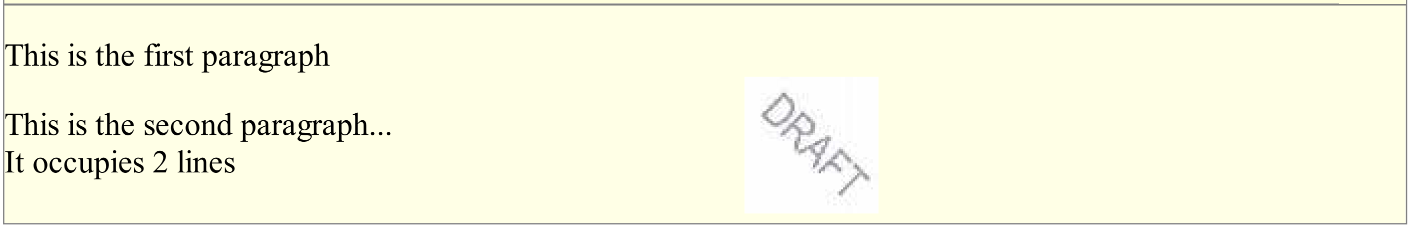
This is the second paragraph...
It occupies 2 lines

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

background-position

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

CSS



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

CSS

W3C CSS Validator (3.2.6)

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

CSS

output

- 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