CSE3026: Web Application Development
CSS for Styling

Scott Uk-Jin Lee

3.1: Basic CSS

- **3.1: Basic CSS**
- 3.2: CSS Properties
- 3.3: More CSS Syntax

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>
```

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

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

Cascading Style Sheets (CSS): `<link>`

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

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

- **CSS** describes the appearance and layout of information on a web page
  - (as opposed to HTML, which describes the content of the page)
- can be embedded in HTML or placed into separate .css file (preferred)

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**
- a rule's **selector** specifies HTML element(s) and applies style **properties**
  - a selector of * selects all elements

CSS properties for colors

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

This paragraph uses the style above.

property	description
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 h2 uses the second style above.

This h4 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**

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";
}
```

This paragraph uses the first style above.

This h2 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.

- 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;
}
```

This paragraph uses the style above.

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

font

```
p {
  font: italic bold 14px "Arial", cursive;
}
```

This paragraph uses the style above.

- can set many aspects of the fint in one step (short cut property)

Recall: Basic CSS rule syntax

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

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

Grouping styles

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

This paragraph uses the above style.

This h2 uses the above styles.

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

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 much as code such as Java
- the // single-line comment style is NOT supported in CSS
- the <!-- ... --> HTML comment style is also NOT supported in CSS

W3C CSS Validator

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

3.2: CSS Properties

- 3.1: Basic CSS
- **3.2: CSS Properties**
- 3.3: More CSS Syntax

CSS properties for text

property	description
text-align	alignment of text within its element
text-decoration	decorations such as underlining
text-indent	indents the first letter of each paragraph
text-shadow	a colored shadow near an existing piece of text 🆔
line-height, word-spacing, letter-spacing	gaps between the various portions of the text
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.

- 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, or none
- effects can be combined:

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

text-shadow 🛡️

```
p {
  font-weight: bold;
  text-shadow: -2px 5px gray;
}
```

This paragraph uses the style above.

- shadow is specified as an X-offset, a Y-offset, and an optional color

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

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

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;
}
```

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;
}
```

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

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`

Styling tables

```
table { border: 2px solid black; caption-side: bottom; }
tr { font-style: italic; }
td { background-color: yellow; text-align: center; width: 30%; }
```

Column 1	Column 2
1,1	1,2 okay
2,1 real wide	2,2
My important data	

- all standard CSS styles can be applied to a table, row, or cell
- table specific CSS properties:
    - `border-collapse`, `border-spacing`, `caption-side`, `empty-cells`, `table-layout`

The `border-collapse` property

```
table, td, th { border: 2px solid black; }
table { border-collapse: collapse; }
```

Without border-
collapse

Column 1	Column 2
1,1	1,2
2,1	2,2

With border-
collapse

Column 1	Column 2
1,1	1,2
2,1	2,2

- by default, the overall table has a separate border from each cell inside
- the `border-collapse` property merges these borders into one

The `rowspan` and `colspan` attributes

```
<table>
 <tr><th>Column 1</th><th>Column 2</th><th>Column 3</th></tr>
 <tr><td colspan="2">1,1-1,2</td>
   <td rowspan="3">1,3-3,3</td></tr>
 <tr><td>2,1</td><td>2,2</td></tr>
 <tr><td>3,1</td><td>3,2</td></tr>
</table>
```

Column 1	Column 2	Column 3
1,1-1,2		
2,1	2,2	1,3-3,3
3,1	3,2	

- `colspan` makes a cell occupy multiple columns; `rowspan` multiple rows
- `text-align` and `vertical-align` control where the text appears within a cell

Column styles: `<col>`, `<colgroup>`

```
<table>
 <col class="urgent" />
 <colgroup class="highlight" span="2"></colgroup>

 <tr><th>Column 1</th><th>Column 2</th><th>Column 3</th></tr>
 <tr><td>1,1</td><td>1,2</td><td>1,3</td></tr>
 <tr><td>2,1</td><td>2,2</td><td>2,3</td></tr>
</table>
```

Column 1	Column 2	Column 3
1,1	1,2	1,3
2,1	2,2	2,3

- `col` tag can be used to define styles that apply to an entire column (self-closing)
- `colgroup` tag applies a style to a group of columns (NOT self-closing)

Don't use tables for layout!

- (borderless) tables appear to be an easy way to achieve grid-like page layouts
  - many "newbie" web pages do this
- but, a `table` has semantics; it should be used only to represent an actual table of data
- instead of tables, use `divs`, widths/margins, floats, etc. to perform layout

- tables should not be used for layout!
- Tables should not be used for layout!!
- TABLES SHOULD NOT BE USED FOR LAYOUT!!!
- **TABLES** SHOULD **NOT** BE USED FOR **LAYOUT**!!!!

CSS 3 **new features**

- new **selectors**: `nth-child`, `inline-block`, `:not`, `+`
- ability to **embed fonts** in a page *(yay)*
- easy built-in support for **multi-column layouts**

T his page demonstrates **CSS3** multi-columns, rounded corners, text & box shadows, HSL/HSLA colour selection, nth-*	be floated or positioned (□▸Bug 238072). Column text flow is much improved and border-radius is now antialiased.	of the CSS2 or CSS3 specifically tested. of support for position:fixed and PNG a transparency renders the header and fo

- transparency/**opacity**, color **gradients**, **shadows**
- **rounded corners**/borders
- **animations** and transitions (like Scriptaculous)
- affine **transformations** (scaling, rotation, perspective)

3.3: More CSS Syntax

- 3.1: Basic CSS
- 3.2: CSS Properties
- **3.3: More CSS Syntax**

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

Styles that conflict

```
body { color: green; }
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
- (later we will learn about more specific styles that can override more general styles)

Embedding style sheets: `<style>` (BAD!)

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

- CSS code can be embedded within the head of an HTML page
- this is *bad style*; DO NOT DO THIS (why?)

Inline styles: the `style` attribute (BAD!)

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

This is a paragraph

- higher precedence than embedded or linked styles
- used for one-time overrides and styling a particular element
- this is *bad style*; DO NOT DO THIS (why?)

Content vs. presentation

- HTML is for **content**; *what* is on the page (heading; list; code; etc.)
- CSS is for **presentation**; how to display the page (bold; centered; 20px margin; etc.)
- keeping content separate from presentation is a very important web design principle
- If the HTML contains no styles, its entire appearance can be changed by swapping .css files
- see also: CSS Zen Garden

Cascading style sheets

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

Inheriting styles

```
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; }
```

<div style="text-align:center">This is a heading.</div>

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)

IDs & ID selectors

```
#id {
    property: value;
    property: value;
    ...
    property: value;
}
```

```
<h2 id="asia">Korea</h2>
```

```
#asia {
    font-style: italic;
}
```

Korea

- each ID must be unique throughout the HTML document
- the value of ID attribute begins with a letter, followed by letters, digits, hyphens, underscore, colons, and periods
- ID can also be used in HTML to link to a section of a page

Classes & Class selectors

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

```
<p>first paragraph</p>
<p class="important">second paragraph</p>
<p class="new">third paragraph</p>
<p class="important new">fourth paragraph</p>
```

```
.important { color: red; }
.new { background-color: yellow; }
```

first paragraph

second paragraph

third paragraph

fourth paragraph

- class attribute is an identifier you can attach to any HTML element
- multiple elements can have the same class value

CSS pseudo-classes

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

Check out the CSE dept. webpage!

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
:nth-child(N)	applies to every Nth child of a given parent