CSS

Class 4

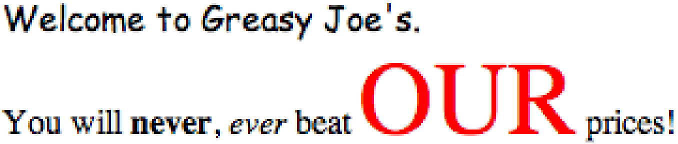
Separation of Duties

Reprise:

- a fundamental concept of modern systems is the separation of:
  - content
  - appearance
  - behavior
- appearance is provided by Cascading Style Sheets (CSS)

Content-Based Appearance

```
<p>
  <font face="Comic Sans MS">Welcome to Greasy Joe’s.
  </font>You will <b>never</b>, <i>ever</i> beat
  <font size="+4" color="red">OUR</font> prices!
</p>
```



Welcome to Greasy Joe's.

You will **never**, *ever* beat OUR prices!

CSS

- CSS 1: 1996
- CSS 2: 1998
- CSS 2.1: 2004 – 2011
  - fully supported by all modern browsers
- CSS 3: 1999 – present
  - split into modules
  - many core modules fully implemented
  - some modules still under development
  - almost complete modern browser support
- CSS 4: 2007 – present
  - split into modules
  - some modules fully implemented
  - most modules still under development
  - some browser support

CSS Locations

- CSS can be placed:
  - inline within HTML body: terrible, conflates html and css
    never do this
  - in the HTML header
    - ok for testing
    - ok if you need a document entirely in a single file
    - not ok for production web pages
      and never in this class
  - in a separate .css file: full separation, reusable
    we will always do this

CSS Syntax

```
selector(s)
{
  property: value;
  property: value;
  ...
  property: value;
}
```

- a CSS file consists of rules
- a rule's selectors match HTML elements
- a rule's properties are applied to the matched elements

Applying CSS to a Page

include a link tag in the page's header

```
<head>
  <meta charset="utf-8" />
  <meta name="author" content="Jon Beck" />
  <link rel="stylesheet" href="315.css" />
  <title>CS 315 Internet Programming</title>
</head>
```

Relative vs Absolute Links

Consider the page

`http://sand.truman.edu/~jbeck/119/foo.html`

that contains the following:

```
...
<link rel="stylesheet" href="foo.css" />
<link rel="stylesheet" href="/foo.css" />
...
<img src="foo.png" alt="foo" />
<img src="../images/foo.png" alt="foo" />
<img src="http://sand.truman.edu/~jbeck/119/foo.png" alt="foo" />
<img src="http://imgs.xkcd.com/comics/ceres.png" alt="foo" />
```

Color

CSS can be used to specify color using various color schemes:

- 140 predefined names
  `https://developer.mozilla.org/en-US/docs/Web/CSS/color_value`
  - red
  - mediumspringgreen
- $2^{24} = 16\text{Mi}$ rgb color codes
  - rgb(255, 0, 0)
- 2^{24} equivalent rgb hex color codes
  - #FF0000: the most common system
- rgba (rgb plus alpha to specify opacity)
  - rgba(255, 0, 0, 0.35)
- hsl (hue, saturation, lightness): sometimes used professionally but hard to understand

- like html character entities, names are preferred
- use numbers only if no name exists

Color Example

color.html
color.css

CSS Font Properties

a font is a named coherent set of character properties

CSS Property	Value
font-family	serif, “Courier New”
font-size	literal, percentage, or named
font-style	normal, italic
font-weight	normal, bold, bolder
font-variant	normal, small-caps

Font Family

- all browsers properly handle these families
  - serif
  - sans-serif
  - monospace
  - fantasy
  - cursive
- but not every browser knows about these font families
  - "Groovy Baby"
  - MissTerious

if you must use fancy fonts, it's safest to use google's fonts at `https://fonts.google.com`

Font Size

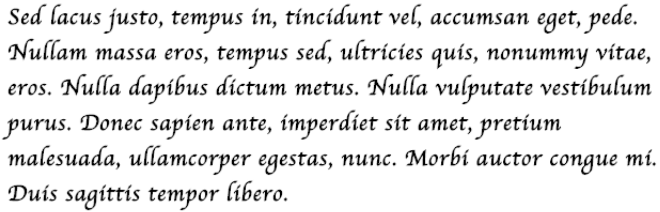
- font size can be specified in
  - absolute units of pixels (px) or points (pt)
  - relative units of uppercase-M's width (em)
  - absolute name: xx-small x-small small medium large x-large xx-large
  - percentage of the parent element's font size
  - named relative to parent element's font size: smaller larger

Font Style and Weight

- font-style
  - normal (default for most elements)
  - italic
- font-weight
  - normal (default for most elements)
  - bold (default for h1 etc.)

Font Properties Example

```
p
{
  font-style: italic;
  font-weight: bold;
  font-size: 14px;
  font-family: cursive;
}
```



Sed lacus justo, tempus in, tincidunt vel, accumsan eget, pede. Nullam massa eros, tempus sed, ultricies quis, nonummy vitae, eros. Nulla dapibus dictum metus. Nulla vulputate vestibulum purus. Donec sapien ante, imperdiet sit amet, pretium malesuada, ullamcorper egestas, nunc. Morbi auctor congue mi. Duis sagittis tempor libero.

Assignment 118

- meta, br, hr, and img are self-closing tags
- no space on either side of = on tag attributes
  no: `foo = "bar"`
  yes: `foo="bar"`
- h2 with no h1
- line length
- non-tabular stuff in a table: list
- img, a, and text are inline elements, and so can only exist inside of a block element
- indenting issues: 2 spaces, no tabs

Assignment 118

- relative links for resources on sand

Logical Elements

consider the sentence:

Now is the time for *all* good men to come to the aid of their country.

what is the way in a web page to make the letters of "all" appear in italics this way?

Semantics

A key part of the separation of duties is the concept of logically defining an element's role in HTML, and using CSS to dictate the appearance of that element.

How do we logically define “all” as an element to be emphasized?

NOT this way:

```
<p>
  Now is the time for <i>all</i> good men to come
  to the aid of their country.
</p>
```

- because HTML is for content, not for appearance

Logical Element Definition

This is the right way:

```
<p>
  Now is the time for <em>all</em> good men to come to the
  aid of their country.
</p>
```

Similar Elements

- beginning developers are often confused by tags that seem to produce identical results
- all of these elements produce “italic” text:
  - ⟨i⟩
  - ⟨em⟩
  - ⟨cite⟩
  - ⟨dfn⟩
- which one to use?

Logical Elements

i the idiomatic element for taxonomy (genus and species), foreign lnaguage phrase (*Et tu, Brute*), technical terms, name of a ship (*Queen Mary*)

em the emphasis element for voice stress emphasis (So just *leave* already.)

cite the citation element for the name of a work, such as a book, play, or song

dfn the definition element for the defining use of a term in text (similar to dt-dd pair, but inline text instead of a list)

strong the strong importance element, is for a section of text that is important, even though you wouldn’t normally stress it by voice (Caution: **The floor may be slippery!**)

Browser Rendering

- by default, browsers render ⟨i⟩, ⟨em⟩, ⟨cite⟩, and ⟨dfn⟩ with italic font
- but what if you want something different?
- for example, you want definitions underlined unstead of italicized?
- then you use CSS to override the browser defaults

```
dfn
{
  font-style: normal;
  text-decoration: underline;
}
```

dfn_underline.html
dfn_underline.css

Text Properties

CSS Property	Value
text-align	left, center, right, justify
text-decoration	underline, overline, line-through, etc
text-indent	indent first line using units of px, pt, %, or em
text-shadow	drop-shadow: two distances plus shadow color
line-height	a size in units of px, pt, %, or em
letter-spacing	a size in units of px, pt, %, or em
word-spacing	a size in units of px, pt, %, or em

text.html

text.css

Background Properties

CSS Property	Value
background-color	a color
background-image	url(image location)
background-repeat	repeat, repeat-x, repeat-y, no-repeat

background.html
background.css

List Property

Each type of list has a default appearance, but you can change a list's list-style-type property. Values include:

- none
- disc (filled)
- circle (hollow)
- square (filled)
- decimal
- decimal-leading-zero
- lower-roman
- upper-roman
- lower-alpha
- lower-greek
- hebrew

List Property Example

There's more to styling lists, that we'll get to eventually.

list.html
list.css

Inheritance

- most styles rules apply not only to an element but also to the elements nested inside it
- the inner element inherits the style of its parent

```
<body>
  <p>Hello, world.</p>
</body>

body
{
  text-decoration: underline;
  color: red;
}
```

Conflicts

- multiple rules can match the same element
- two rules can conflict
  - the most specific or closely matching selector wins
  - if a tie, the last selector wins

```
body
{
  background-color: cyan;
}

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

```
h2
{
  color: red;
}

h2
{
  color: blue;
}
```

Selectors

- a selector that lists an element type matches all such elements

```
p
{
  color: red;
}
```

Class Selectors

- a selector can specify a class of elements
- the class name is an identifier you make up

```
<p>Normal paragraph.</p>
<p class="example">Paragraph of a specific class.</p>

p.example
{
  color: red;
}
```

Class Selectors

- a class selector can specify elements of that class:

```
p.example, h1.example
{
  color: red;
}
```

- or all elements of that class

```
.example
{
  color: red;
}
```

Class Names

To reiterate:

- appearance should depend on semantics
- choose class names by semantics or purpose, not appearance

Bad Class Name:	`<p class="red">`
Good Class Name:	`<p class="alert">`

Bad Class Name:	`<p class="centered">`
Good Class Name:	`<p class="quotation">`

ID Selectors

- a selector can specify the ID of an element
- the ID name is an identifier you make up
- the ID must be unique on the entire page

```
<p id="first">The first paragraph.</p>
<p>All other paragraphs.</p>

#first
{
  color: red;
}

p
{
  color: blue;
}
```

Elements in Multiple Classes

- an element can belong to multiple classes

```
<p class="exam special">Belongs to multiple classes.</p>

.exam
{
  color: blue;
}

.special
{
  background-color: red;
}
```

- if multiple class rules conflict, the last one listed wins

Pseudo-class Selectors

- a pseudo-class targets an element based on context or state
- pseudo-classes are pre-defined
- pseudo-class selectors begin with a colon
- pseudo-classes are not identified in the HTML file

Pseudo-class	**Value**
:visited, :link	links that have or have not been clicked
:checked	a checkbox that is currently checked
:focus	input element that has the user input focus
:hover	element that has the mouse pointer on it
:nth-child(expr)	specific children of a given element
:not(selector)	all elements that do not match the selector
::first-line	the first line of text within a block (2 colons)

Pseudo-Class Selector Example

pseudo-class.html
pseudo-class.css

CSS Validation

- CSS is validated similar to HTML
- `https://jigsaw.w3.org/css-validator`