

CSc 337



LECTURE 2: MORE HTML AND CSS

Activity: match this page

Koala Bears

Koalas are **marsupials** and **not actually bears**.

Food

Koalas eat eucalyptus.

Dangers

(Koalas are an endangered species)

Dingos & Deforestation & Roads & Disease.

Pictures



('O')

Page Text:

Koala Bears

Koalas are marsupials and not actually bears.

Food

Koalas eat eucalyptus.

Dangers

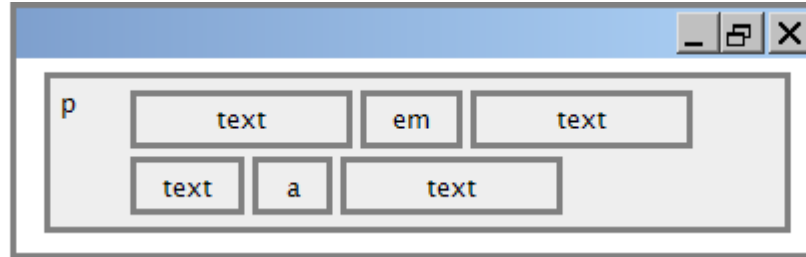
(Koalas are an endangered species)

Dingos & Deforestation & Roads & Disease.

Pictures

('O')

Block and inline elements



block elements contain an entire large region of content

- examples: paragraphs, lists, table cells
- the browser places a margin of whitespace between block elements for separation

inline elements affect a small amount of content

- examples: bold text, code fragments, images
- the browser allows many inline elements to appear on the same line
- must be nested inside a block element

Nesting tags

```
<p>  
    HTML is <em>really,  
    <strong>REALLY</em> lots of</strong> fun!  
</p>
```

- tags must be correctly nested
 - (a closing tag must match the most recently opened tag)
- the browser may render it correctly anyway, but it is invalid HTML
 - (how would we get the above effect in a valid way?)

Comments: `<!-- ... -->`

comments to document your HTML file or "comment out" text

```
<!-- My web page, by Suzy Student  
      CSE 190 D, Spring 2048      -->  
<p>CSE courses are <!-- NOT --> a lot of fun!</p>
```

CSE courses are a lot of fun!

- many web pages are not thoroughly commented (or at all)
- still useful at top of page and for disabling code
- comments cannot be nested and cannot contain a --

Unordered list: ,

```
<ul>  
  <li>No shoes</li>  
  <li>No shirt</li>  
  <li>No problem!</li>  
</ul>
```

HTML

- No shoes
- No shirt
- No problem!

output

- **ul** represents a bulleted list of items (block)
- **li** represents a single item within the list (block)

More about unordered lists

```
<ul>
  <li>Harry Potter characters:
    <ul>
      <li>Harry Potter</li>
      <li>Hermione</li>
      <li>Ron</li>
    </ul>
  </li>
  <li>LOTR characters:
    <ul>
      <li>Frodo</li>
      <li>Bilbo</li>
      <li>Sam</li>
    </ul>
  </li>
</ul>
```

HTML

- Harry Potter characters:
 - Harry Potter
 - Hermione
 - Ron
- LOTR characters:
 - Frodo
 - Bilbo
 - Sam

output

Ordered list

```
<p>Apple business model:</p>
<ol>
  <li>Beat Microsoft</li>
  <li>Beat Google</li>
  <li>Conquer the world!</li>
</ol>
```

HTML

Apple business model:

1. Beat Microsoft
2. Beat Google
3. Conquer the world

output

- **ol** represents a numbered list of items
- we can make lists with letters or Roman numerals using CSS (later)

Definition list <dl>, <dt>, <dd>

```
<dl>  
  <dt>newbie</dt> <dd>one who does not have mad skills</dd>  
  <dt>own</dt> <dd>to soundly defeat (e.g. I owned that newbie!)</dd>  
  <dt>frag</dt> <dd>a kill in a shooting game</dd>  
</dl>
```

HTML

newbie
one who does not have mad skills

own
to soundly defeat (e.g. I owned that newbie!)

frag
a kill in a shooting game

output

- **dl** represents a list of definitions of terms
- **dt** represents each term, and **dd** its definition

Web Standards

It is important to write proper HTML code and follow proper syntax.

Why use valid HTML and web standards?

- more rigid and structured language
- more interoperable across different web browsers
- more likely that our pages will display correctly in the future
- can be interchanged with other XML data: [SVG](#) (graphics), [MathML](#), [MusicML](#), [etc.](#)

W3C HTML Validator

```
<p>  
  <a href="http://validator.w3.org/check/referer">  
      
  </a>  
</p>
```



- validator.w3.org
- checks your HTML code to make sure it follows the official HTML syntax
- more picky than the browser, which may render bad HTML correctly

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

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

HTML

- 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

Inline styles: the style attribute (BAD!)

```
<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*; DO NOT DO THIS (why?)

Embedding style sheets: <style> (BAD!)

```
<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*; 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](#)

Activity: match this page

Koala ~~Bears~~

Koalas are marsupials and not actually bears.

Food

Koalas eat eucalyptus.

Dangers

(Koalas are an endangered species)

Dingos & Deforestation & Roads & Disease.

Pictures



('0')

The headings are Verdana, Geneva, Arial or any sans-serif font on the machine.

The overall heading is 5ems big, not bold and has a 5 by 5 green shadow.

CSS properties for colors

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

This paragraph uses the style above.

Property	Description
<u>color</u>	color of an element's text
<u>background-color</u>	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), 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
<u>font-family</u>	which font will be used
<u>font-size</u>	how large the letters will be drawn
<u>font-style</u>	used to enable/disable italic style
<u>font-weight</u>	used to enable/disable bold style
<u>Complete list of font properties</u>	

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%

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

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)

CSS properties for text

property	description
<code>text-align</code>	alignment of text within its element
<code>text-decoration</code>	decorations such as underlining
<code>line-height</code> , <code>word-spacing</code> , <code>letter-spacing</code>	gaps between the various portions of the text
<code>text-indent</code>	indents the first letter of each paragraph

[Complete list of text properties](http://www.w3schools.com/css/css_reference.asp#text) (http://www.w3schools.com/css/css_reference.asp#text)

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

- 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;
```

text-shadow

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

CSS

This paragraph uses the style above.

output

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

The list-style-type property

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

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");  
}
```

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

The visibility property

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

CSS

output

property	description
visibility	sets whether an element should be shown onscreen; can be visible (default) or hidden

- `hidden` elements 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 used to show/hide dynamic HTML content on the page in response to events

The opacity property

```
body { background-image: url("images/marty-mcfly.jpg");  
background-repeat: repeat; }  
p { background-color: yellow; }  
p.mcfly1 { opacity: 0.75; }  
p.mcfly2 { opacity: 0.50; }  
p.mcfly3 { opacity: 0.25; }
```

CSS

Marty McFly in 1985

Marty McFly in 1955 fading away, stage 1

Marty McFly in 1955 fading away, stage 2

Marty McFly in 1955 fading away, stage 3



property	description
opacity	how not-transparent the element is; value ranges from 1.0 (opaque) to 0.0 (transparent)

box-shadow

```
box-shadow: h-shadow v-shadow blur;
```

CSS

```
box-shadow: 10px 10px 5px;
```

CSS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam vitae viverra nulla, sit amet vulputate elit. Curabitur sit amet augue venenatis, facilisis nulla ac, aliquet erat. Nullam diam nibh, pharetra ut mi eget, efficitur aliquam ante. Nunc non ipsum a turpis ultrices ultrices. Donec ut massa euismod, egestas diam non, rutrum massa. Ut id risus et nibh scelerisque porta. Quisque volutpat rhoncus tellus. Aenean mollis commodo urna. Nunc magna sapien, interdum nec arcu id, rhoncus gravida urna. Suspendisse ex odio, consequat eu lorem vestibulum, volutpat sollicitudin nulla. Aenean ac libero velit. Proin consequat augue mi, sit amet consequat dui hendrerit et. Ut eleifend, tellus quis gravida facilisis, neque ante hendrerit magna, quis rhoncus est lorem eu felis.

Grouping styles

```
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

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

```
/* 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 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="http://jigsaw.w3.org/css-validator/check/referer">  
    </a>  
</p>
```

HTML



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