

CSE 154

LECTURE 1: BASIC HTML AND CSS

Hypertext Markup Language (HTML)

- describes the *content* and structure of information on a web page
 - not the same as the *presentation* (appearance on screen)
- surrounds text content with opening and closing tags
- each tag's name is called an element
 - syntax: `<element> content </element>`
 - example: `<p>This is a paragraph</p>`
- most whitespace is insignificant in HTML (ignored or collapsed to a single space)
- we will use a newer version called HTML5

Structure of an HTML page

```
<!DOCTYPE html>
<html>
  <head>
    information about the page
  </head>
  <body>
    page contents
  </body>
</html>
```

Page title: `<title>`

describes the title of the web page

`<title>`Chapter 2: HTML Basics`</title>`

- placed within the head of the page
- displayed in the web browser's title bar and when bookmarking the page

Paragraph: `<p>`

paragraphs of text (block)

```
<p>You're not your job.  
You're not how much money you have in the bank.  
You're not the car you drive.    You're not the contents  
of your wallet. You're not your          khakis.  You're  
the all-singing, all-dancing crap of the world.</p>
```

You're not your job. You're not how much money you have in the bank. You're not the car you drive. You're not the contents of your wallet. You're not your khakis. You're the all-singing, all-dancing crap of the world.

- placed within the body of the page
- [more paragraph examples](#)

Headings: `<h1>`, `<h2>`, ..., `<h6>`

headings to separate major areas of the page (block)

```
<h1>University of Whoville</h1>
```

```
<h2>Department of Computer Science</h2>
```

```
<h3>Sponsored by Micro$oft</h3>
```

University of Whoville

Department of Computer Science

Sponsored by Micro\$oft

- [More heading examples](#)

More about HTML tags

- some tags can contain additional information called attributes
 - syntax: `<element attribute="value" attribute="value"> content </element>`
 - example: `Next page`
- some tags don't contain content; can be opened and closed in one tag
 - syntax: `<element attribute="value" attribute="value" />`
 - example: `<hr />`
 - example: ``

Horizontal rule: `<hr>`

a horizontal line to visually separate sections of a page (block)

```
<p>First paragraph</p>  
<hr />  
<p>Second paragraph</p>
```

First paragraph

Second paragraph

- should be immediately closed with `/>`

Links: `<a>`

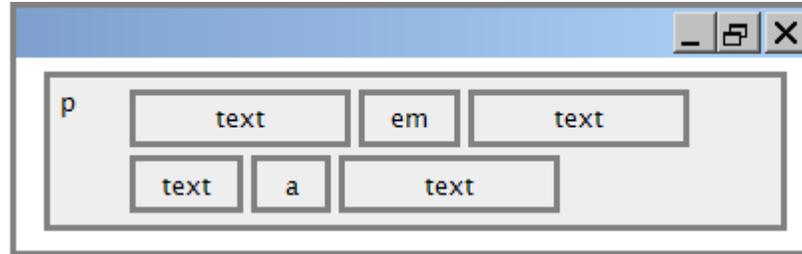
links, or "anchors", to other pages (inline)

```
<p>  
  Search  
  <a href="http://www.google.com/">Google</a> or our  
  <a href="lectures.html">Lecture Notes</a>.  
</p>
```

Search Google or our Lecture Notes.

- uses the href attribute to specify the destination URL
 - can be absolute (to another web site) or relative (to another page on this site)
- anchors are inline elements; must be placed in a block element such as p or h1

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

Images: ``

inserts a graphical image into the page (inline)

```

```



- the src attribute specifies the image URL
- HTML5 also requires an alt attribute describing the image

Line break: `
`

forces a line break in the middle of a block element (inline)

```
<p>The woods are lovely, dark and deep, <br /> But I have promises to keep,
```

```
<br /> And miles to go before I sleep, <br /> And miles to go before I sleep.</p>
```

The woods are lovely, dark and deep,
But I have promises to keep,
And miles to go before I sleep,
And miles to go before I sleep.

- Warning: Don't over-use br (guideline: ≥ 2 in a row is bad)

Phrase elements : ``, ``

em: emphasized text (usually rendered in italic)

strong: strongly emphasized text (usually rendered in bold)

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

HTML is *really*, **REALLY** fun!

- as usual, the tags must be properly nested for a valid page

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

Quotations <blockquote>

a lengthy quotation (block)

```
<p>As Lincoln said in his famous Gettysburg Address:</p>
  <blockquote>
    <p>Fourscore and seven years ago, our fathers brought forth
    on this continent a new nation, conceived in liberty, and
    dedicated to the proposition that all men are created equal.</p>
  </blockquote>
```

HTML

As Lincoln said in his famous Gettysburg Address:

Fourscore and seven years ago, our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

output

Inline quotations <q>

a short quotation (inline)

| | |
|--|--------|
| <code><p>Quoth the Raven, <q>Nevermore.</q></p></code> | HTML |
| Quoth the Raven, “Nevermore.” | output |

- Why not just write the following?

`<p>Quoth the Raven, "Nevermore."</p>`

Deletions and insertions: ``, `<ins>`

content that should be considered deleted or added to the document (inline)

```
<p>  
  <del>Final Exam</del> <ins>Midterm</ins> is on  
  <del>Aug 29</del> <ins>Apr 17</ins>.  
</p>
```

HTML

~~Final Exam~~ Midterm is on ~~Aug 29~~ Apr 17.

output

Abbreviations: <abbr>

an abbreviation, acronym, or slang term (inline)

```
<p>
```

```
Safe divers always remember to check their  
<abbr title="Self-Contained Underwater Breathing  
Apparatus">SCUBA</abbr> gear.
```

```
</p>
```

HTML

Safe divers always remember to check their SCUBA gear.

output

- The title will appear when the abbreviated word is hovered over
- In some browsers the abbreviated word will have a dashed underline

Computer code `<code>`

a short section of computer code (usually shown in a fixed-width font)

```
<p>  
  The ul and ol  
  tags make lists.  
</p>
```

HTML

The `ul` and `ol` tags make lists.

output

Preformatted text <pre>

a large section of pre-formatted text (block)

| | |
|--|--------|
| <pre><pre> Bill Gates speaks You will be assimilated Microsoft fans delirious </pre></pre> | HTML |
| <pre> Bill Gates speaks You will be assimilated Microsoft fans delirious</pre> | output |

- Displayed with exactly the whitespace / line breaks given in the text
- Shown in a fixed-width font by default

Web page metadata: <meta>

information about your page (for a browser, search engine, etc.)

```
<meta charset="utf-8" />  
<meta name="description" content="Authors' web site for Building Java Programs." />  
<meta name="keywords" content="java, textbook" />
```

HTML

- placed **in the head** section of your HTML page
- meta tags often have both the name and content attributes
 - some meta tags use the http-equiv attribute instead of name
 - the meta tag with charset attribute indicates language/character encodings
- using a meta tag Content-Type stops validator "tentatively valid" warnings

Favorites icon ("favicon")

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

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



output

- the link tag, placed in the head section, attaches another file to the page
 - in this case, an icon to be placed in the browser title bar and bookmarks
- IE6: Doesn't work; must put a file favicon.ico in the root of the web server

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

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*, EVER beat **OUR** prices!

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

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

- 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 |
|-------------------------|---|
| <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-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-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-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)