

# Hypertext Markup Language (HTML)

## Web Engineering Class

Department of Computer Engineering, Junior of first semester 2009

본 내용은 University of Washington, CSE 190 M (Web Programming), Spring 2007 수업에서 사용된 자료입니다.

Except where otherwise noted, the contents of this presentation are © Copyright 2007 Marty Stepp and are licensed under the Creative Commons Attribution 2.5 License.



---

## 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  
(it gets ignored or collapsed into a single space)

---

## More about HTML tags

---

- some tags can contain additional information called attributes
  - syntax: `<element attribute="value" attribute="value"> content </element>`
  - example: `<a href="page2.html">Next page</a>`
- some tags don't contain content; can be opened and closed in one tag
  - syntax: `<element attribute="value" attribute="value" />`
  - example: ``
  - example: `<hr />`

---

## Structure of an HTML page

---

`<html>`

`<head>`

information about the page

</head>

<body>

page contents

</body>

</html>

- a header describes the page and a body contains the page's contents
- an HTML page is saved into a file ending with extension .html
- Example page

---

## XHTML

---

- a newer version of HTML, standardized in 2000
- uses a markup format called XML (XML + HTML = XHTML)
- though the browser will accept some malformed HTML, we'll write "strict" XHTML that complies to the official web standards
- why use XHTML 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.
- a strict XHTML page uses some different syntax and tags

---

## Structure of an XHTML page

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<title>information about the page</title>
```

```
</head>
```

```
<body>
```

```
<p>page contents</p>
```

```
</body>
```

```
</html>
```

---

## Page title: <title>

---

describes the title of the web page

```
<title>Web Engineering: HTML</title>
```

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

---

## Block-level and inline elements (explanation)

---

- block-level elements contain an entire large region of text
  - examples: paragraphs, lists, table cells
  - the browser places a margin of whitespace between some block-level elements for separation
- inline elements affect a small amount of text and must be nested inside of a block-level element
  - examples: bold text, code fragments, images
  - the browser allows many inline elements to appear on the same line or within the same block-level element

---

## Paragraph: <p>

---

paragraphs of text (block-level)

```
<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.

- 
- more paragraph examples

---

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

---

headings to separate major areas of the page (block-level)

```
<h1>Cheju National University</h1>  
<h2>Department of Computer Engineering</h2>  
<h3>Web Engineering</h3>
```

## Cheju National University

### Department of Computer Engineering

#### Web Engineering

---

- [More heading examples](#)
- 

## Anchors: `<a>`

---

anchors denote links to other pages (inline)

```
<p>Search  
<a href="http://www.google.com/">Google</a> now!</p>
```

Search [Google](#) now!

---

- uses the href attribute to specify the destination URL
  - anchors are inline elements, so they must be placed inside a block-level element such as a p or h1
- 

## More about anchors

---

```
<p><a href="1-internet.html">Lecture Notes 1</a></p>  
<p><a href="http://www.google.com/"  
title="Search">Google</a></p>
```

[Lecture Notes 1](#)

[Google](#)

---

- types of URLs that can appear in anchors:
    - absolute (to another web site)
    - relative (to another page on this web site)
  - specify a additional information with the title attribute (some user agent displays title as a tooltip)
  - [more anchor examples](#)
  - to make links that open in new windows, we'll need to learn Javascript (later)
- 

## Nesting tags

---

Bad:

```
<p><a href="1-internet.html">Lecture Notes 1</p>
<p>This text also links to Lecture Notes 1</a></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 XHTML
- 

## Line break: <br>

---

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

```
<p>Teddy said it was a hat,<br />So I put it on.</p>
<p>Now Daddy's sayin',<br />
Where the heck's the toilet plunger gone?</p>
```

Teddy said it was a hat,  
So I put it on.

Now Daddy's sayin',  
Where the heck's the toilet plunger gone?

---

- br should be immediately closed with />
  - br should not be used to separate paragraphs or used multiple times in a row to create spacing
- 

## Horizontal rule: <hr>

---

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

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

First paragraph

---

Second paragraph

---

- should be immediately closed with />

---

## Images: <img>

---

inserts a graphical image into the page (inline)

```
<p></p>
```



- 
- the src attribute specifies the image URL
  - XHTML also requires an alt attribute for alternative text

---

## More about images

---

```
<p><a href="http://theonering.net/">  
</a></p>
```



- 
- if placed inside an a anchor, the image will become a link
  - the title attribute specifies an additional information

---

## Practice Problem

---



Bilbo Baggins ([Wiki](#)) and Frodo Baggins ([Wiki](#))

**Result:**

**Interaction Pane:**

Show Answer

Evaluate

---

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

---

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

```
<!-- My web page, by Suzy Student Web Engineering -->
<p>Web Engineering courses are <!-- NOT --> a lot of fun!</p>
```

Web Engineering courses are a lot of fun!

---

## Unordered list: `<ul>`, `<li>`

---

ul represents a bulleted list of items (block-level)

li represents a single item within the list (block-level)

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

- No shoes
- No shirt
- No problem!

# More about unordered lists

- a list can contain other lists

```
<ul>
  <li>Simpsons:
    <ul>
      <li>Bart</li>
      <li>Lisa</li>
    </ul>
  </li>
  <li>Family Guy:
    <ul>
      <li>Peter</li>
      <li>Lois</li>
    </ul>
  </li>
</ul>
```

- Simpsons:
  - Bart
  - Lisa
- Family Guy:
  - Peter
  - Lois

## Ordered list: <ol>

ol represents a numbered list of items (block-level)

```
<p>RIAA business model:</p>
<ol>
  <li>Sell music on easily copyable discs</li>
  <li>Sue customers for copying music</li>
  <li>???</li>
  <li>Profit!</li>
</ol>
```

RIAA business model:

1. Sell music on easily copyable discs
2. Sue customers for copying music
3. ???
4. Profit!

- we can make lists with letters or Roman numerals using CSS (later)

## Common Error: Not closing a list

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

```
<p>Paragraph after list...</p>
```



- No shoes
- No shirt
- No problem!

Paragraph after list...

- 
- if you leave a list open, subsequent contents will be indented

---

## Common Error: Improper nested list placing

---

```
<ul>
  <li>Simpsons:</li>
    <ul>
      <li>Bart</li>
      <li>Lisa</li>
    </ul>
</li>
  <li>Family Guy:
    <ul>
      <li>Peter</li>
      <li>Lois</li>
    </ul>
</ul>
```

- 
- closing the outer li too early (or not at all) will render correctly in most browsers, but it is incorrect XHTML

---

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

---

dl represents a list of definitions of terms (block-level)

dt represents each term, and dd its definition

```
<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>
```

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

---

## Quotations: `<blockquote>`

---

a lengthy quotation (block-level)

```
<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>
```

As Abraham 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.

---

## Quotations: `<q>`

---

a short quotation (inline)

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

Quoth the Raven, "Nevermore."

- 
- Why not just write the following?  

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

We don't use " marks for two reasons:

1. XHTML shouldn't contain literal quotation mark characters; they should be written as `&quot;`;
2. using `<q>` allows us to apply CSS styles to quotations (seen later)

---

## Phrase elements: `<em>`, `<strong>`, `<code>`

---

`em`: emphasized text (usually rendered in italic)

`strong`: strongly emphasized text (usually rendered in bold)

`code`: a short section of computer code (usually rendered in a fixed-width font)

```
<p>The <code>ul</code> and <code>ol</code> tags make lists.</p>
<p>HTML is <em>really</em>, <strong>REALLY</strong> fun!</p>
```

The ul and ol tags make lists.

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

---

## Preformatted text: `<pre>`

---

a large block of code text in a fixed-width font (block-level)

```
<pre>
public static void main(String[] args) {
    System.out.println("Hello, world!");
}
</pre>

public static void main(String[] args) {
    System.out.println("Hello, world!");
}
```

- 
- unlike most of HTML, pre blocks are displayed with exactly the whitespace and line breaks given in the document
  - how would it look if we had instead enclosed it in code tags?

---

## Practice Problem

---



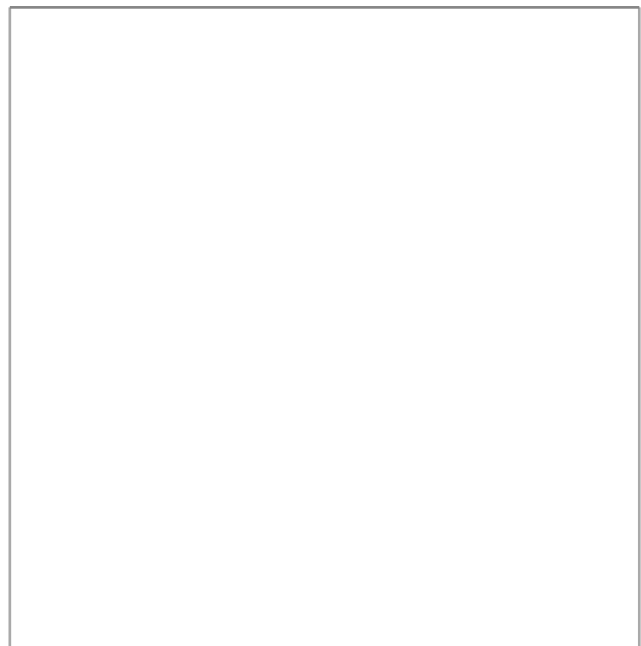
**KITTENS!**



Why I love them:

1. They are *little*.
2. They make **adorable** sounds:
  - "Meow!"
  - "Purr!"
3. JUST LOOK AT THEM!

**Interaction Pane:**



[Show Expected Appearance](#)[Show Answer](#)[Evaluate](#)

---

## W3C XHTML Validator

---

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



- [validator.w3.org](http://validator.w3.org)
- checks your XHTML code to make sure it meets the official strict XHTML specifications
- more picky than the web browser, which may render malformed XHTML correctly

---

## Practice Problem

---

- Combine the earlier HTML code from the hobbit example with the kitten example into a larger complete web page called hobbitten.html.
- Upload the page to a web server.
- Validate the page using the W3C XHTML validator and fix any validation errors that occur.

---

## Web page metadata: <meta>

---

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

```
<meta name="description"  
content="Authors' web site for Building Java Programs." />  
<meta name="keywords" content="java, textbook" />  
<meta http-equiv="Content-Type"  
content="text/html; charset=iso-8859-1" />
```

- placed in the head 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

---

## meta element to describe the page

---

```
<head>
```

```
<meta name="author"
content="web page's author" />
<meta name="revised"
content="web page version and/or last modification date" />
<meta name="generator"
content="the software used to create the page" />
</head>
```

---

- many WYSIWYG HTML editors (FrontPage, PageMaker, etc.) place their names in the meta generator tag (why?)

---

## meta element to aid search engines

---

```
<head>
<meta name="description"
content="how you want search engines to display your page" />
<meta name="keywords"
content="words to associate with your page (comma-separated)" />
</head>
```

---

- these are suggestions to search engines about how to index your page
- the search engine may choose to ignore them (why?)

---

## meta element to aid browser / web server

---

```
<meta http-equiv="refresh"
content="how often to refresh the page (seconds)" />
</head>
<meta http-equiv="Content-Type"
content="what type of document this is
(may include internationalization information such as character encoding)" />
```

---

- using the meta Content-Type tag gets rid of the W3C "tentatively valid" warning  
`<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />`
- the meta refresh tag can also redirect from one page to another:  
`<meta http-equiv="refresh" content="5;url=http://www.bjp.com" />`
  - why would we want to do this? ([example](#))

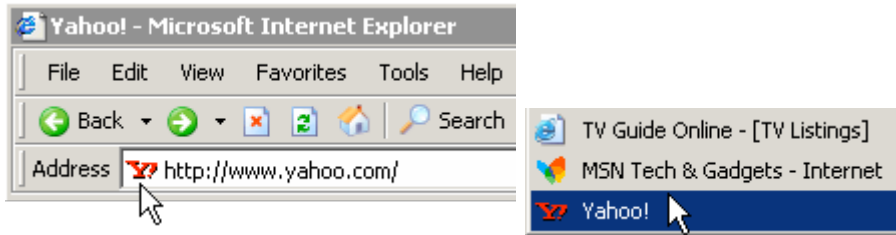
---

## Favorites icon ("favicon")

---

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

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



- 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](#))

## Internet media ("MIME") types

classifications of data that travel over the internet

- sometimes when including resources in a page (style sheet, icon, multimedia object), we specify their type of data

MIME type	file extension
text/html	.html
text/plain	.txt
image/gif	.gif
image/jpeg	.jpg
video/quicktime	.mov
application/octet-stream	.exe

- Lists of MIME types: [by type](#), [by extension](#)

## HTML Character Entities

a way of representing any Unicode character within a web page

character(s)	entity
< >	&lt; &gt;
é è ñ	&eacute; &egrave; &ntilde;
™ ©	&trade; &copy;
π δ Δ	&pi; &delta; &Delta;
И	&#1048;
" &	&quot; &amp;

- Complete list of HTML entities
- How would you display the text & on a web page?

---

## HTML-encoding text

---

<p><a href=  
"http://www.google.com/search?q=marty+stepp&ie=utf-8&aq=t">  
Search Google for Marty</a></p>

- To display the above text in a web page, its special characters must be encoded as shown below:

&lt;p&gt;&lt;a href=  
&quot;http://www.google.com/search?q=marty+stepp&amp;ie=utf-8&amp;aq=t&quot;&gt;  
Search Google for Marty&lt;/a&gt;&lt;/p&gt;