Unit 2

The World Wide Web, Markup and HTML

What is the WWW?

- The World Wide Web is a *service* that operates over the Internet
- Web pages are linked together using hyperlinks
- Creates a vast network of documents and files
- Can think of it as a book
  - With several hundred *billion* pages of which you can access about 11.5 billion
  - The pages can change constantly
  - Multiple authors
  - A book index doesn’t normally change

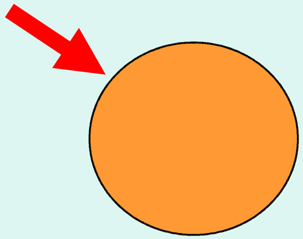
Indexing the WWW

- Finding a page is more difficult than simply typing in a search at www.google.ca
- Google has to be able to find what you’re looking for quickly
  - Can’t search all web pages each time a request is made
  - Wouldn’t even know where to look!
- The answer:
  - Web crawlers, spiders, ants

Web Crawlers

- Web crawlers find web pages and *cache* them for indexing
  - Cache: make a copy of the web page and store it on Search Engine’s servers
- Crawlers start with a list of pre-determined web pages
  - All pages linked from this list are also then added to the list
- Note: when searching on www.google.ca you can choose to view the cached version of the web page

Web Crawler Example

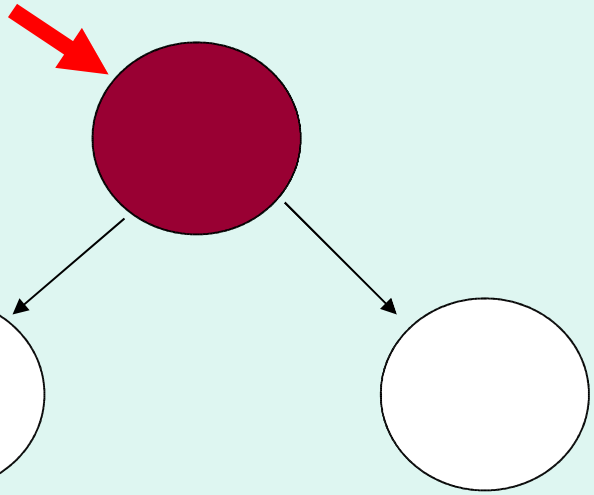


🟧 •On the list of pages to visit and cache

🟥 •Cached Page

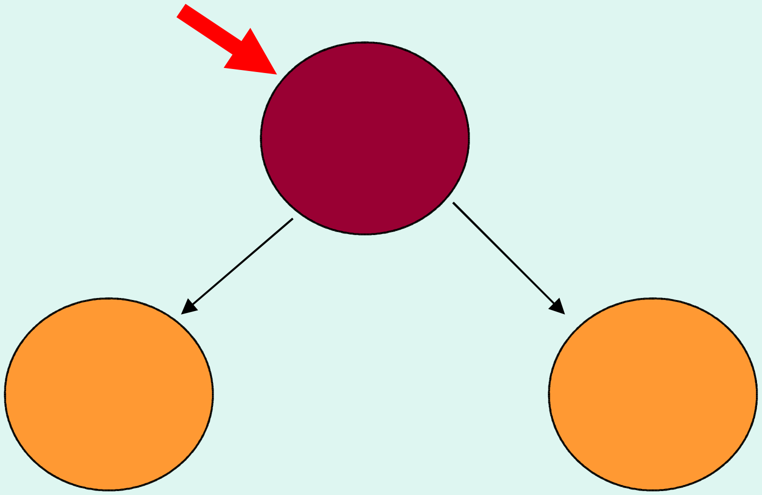
⬜ •Linked Page

Web Crawler Example



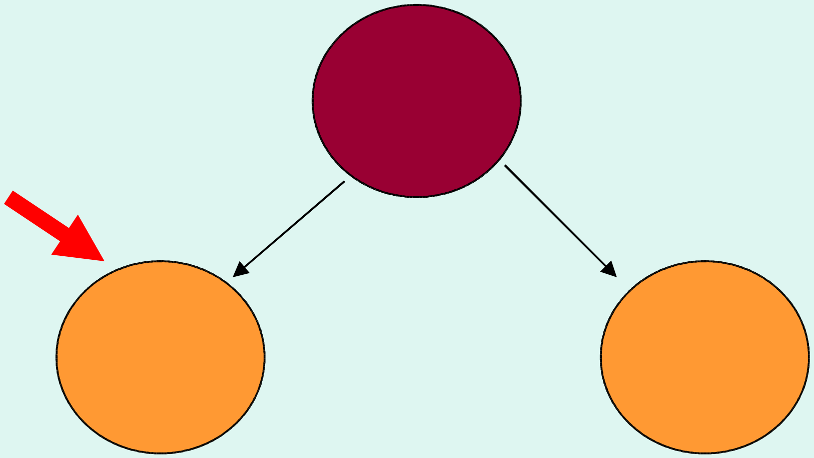
- 🟧 •On the list of pages to visit and cache
- 🟥 •Cached Page
- ⬜ •Linked Page

Web Crawler Example



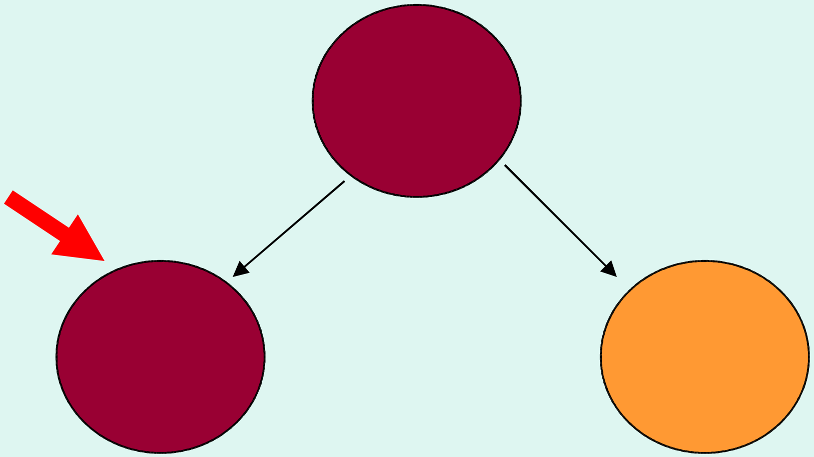
- 🟧 •On the list of pages to visit and cache
- 🟥 •Cached Page
- ⬜ •Linked Page

Web Crawler Example



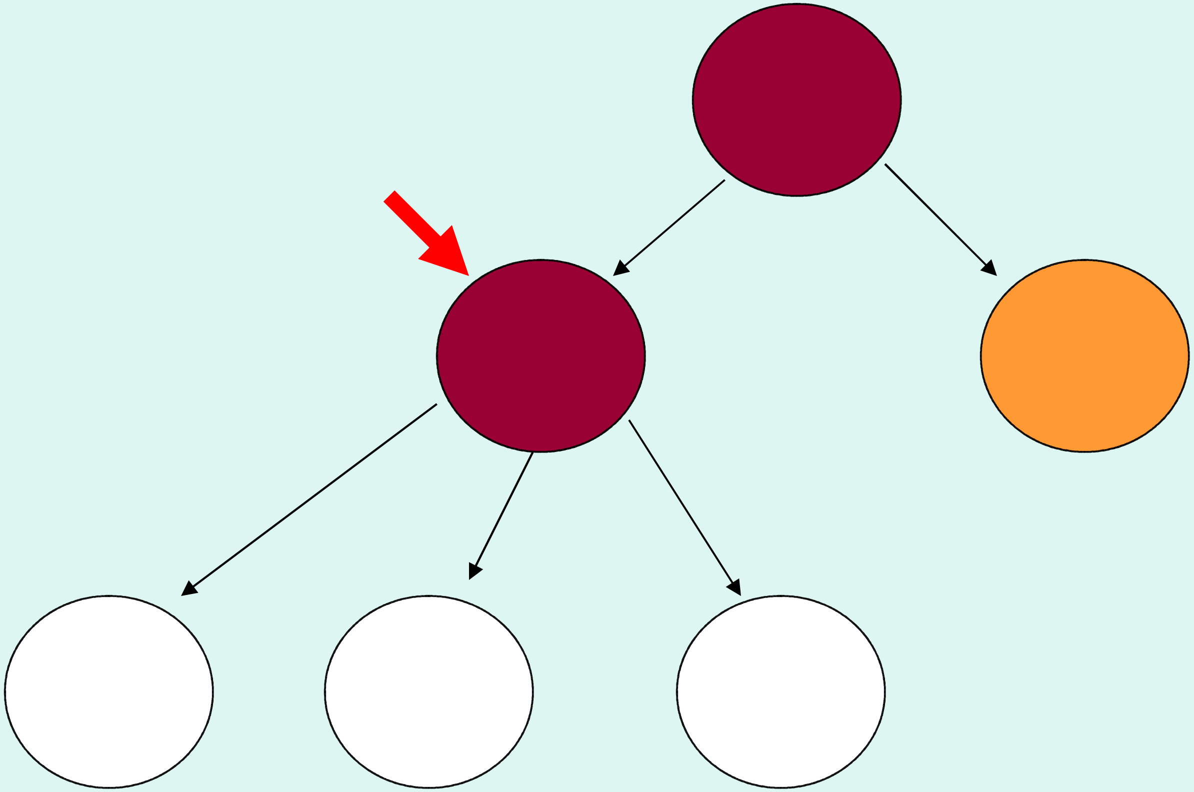
- 🟧 On the list of pages to visit and cache
- 🟥 Cached Page
- ⬜ Linked Page

Web Crawler Example



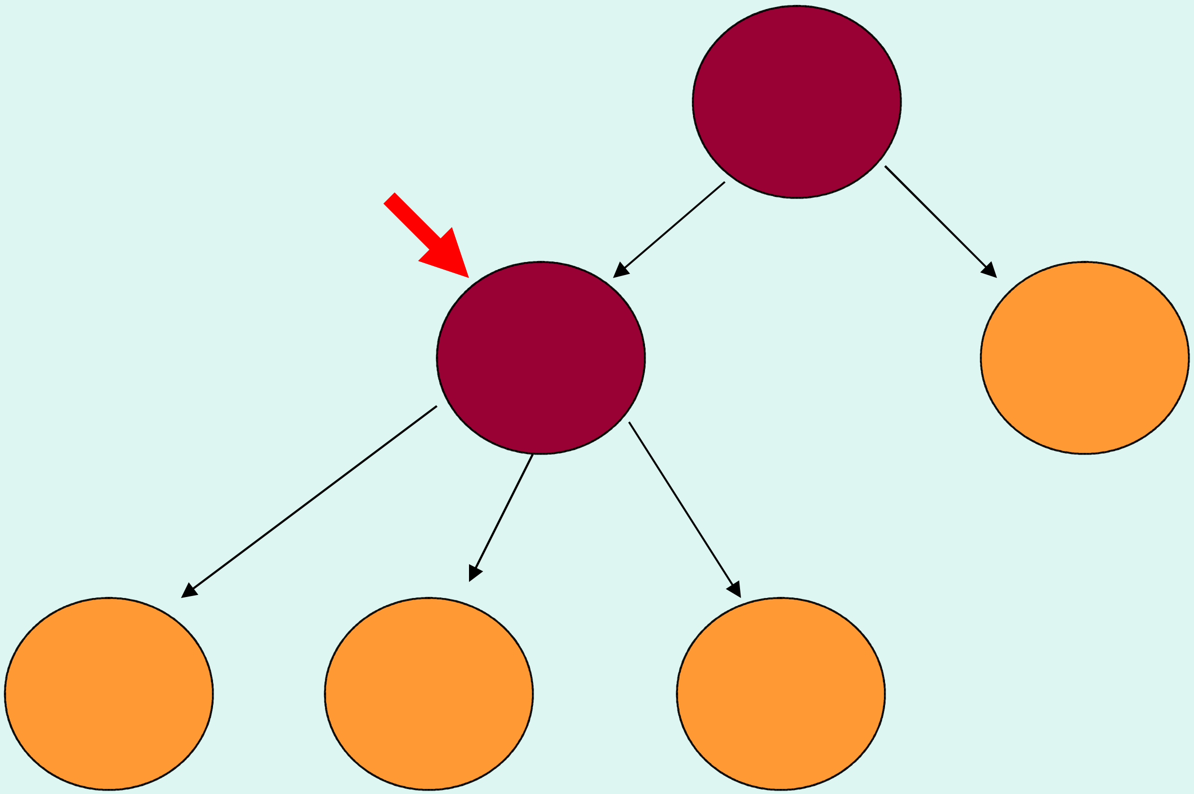
- 🟧 •On the list of pages to visit and cache
- 🟥 •Cached Page
- ⬜ •Linked Page

Web Crawler Example



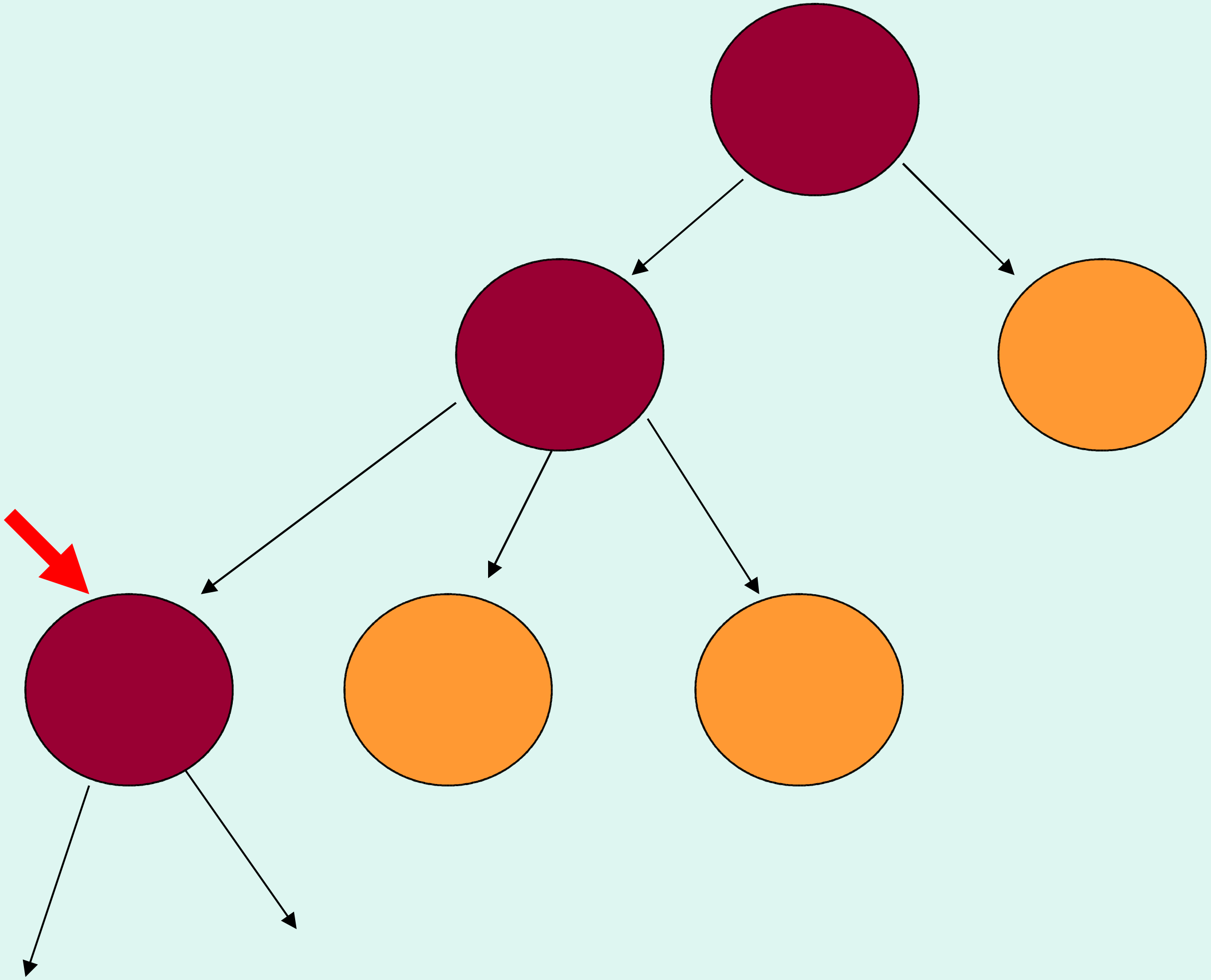
- 🟧 •On the list of pages to visit and cache
- 🟥 •Cached Page
- ⬜ •Linked Page

Web Crawler Example



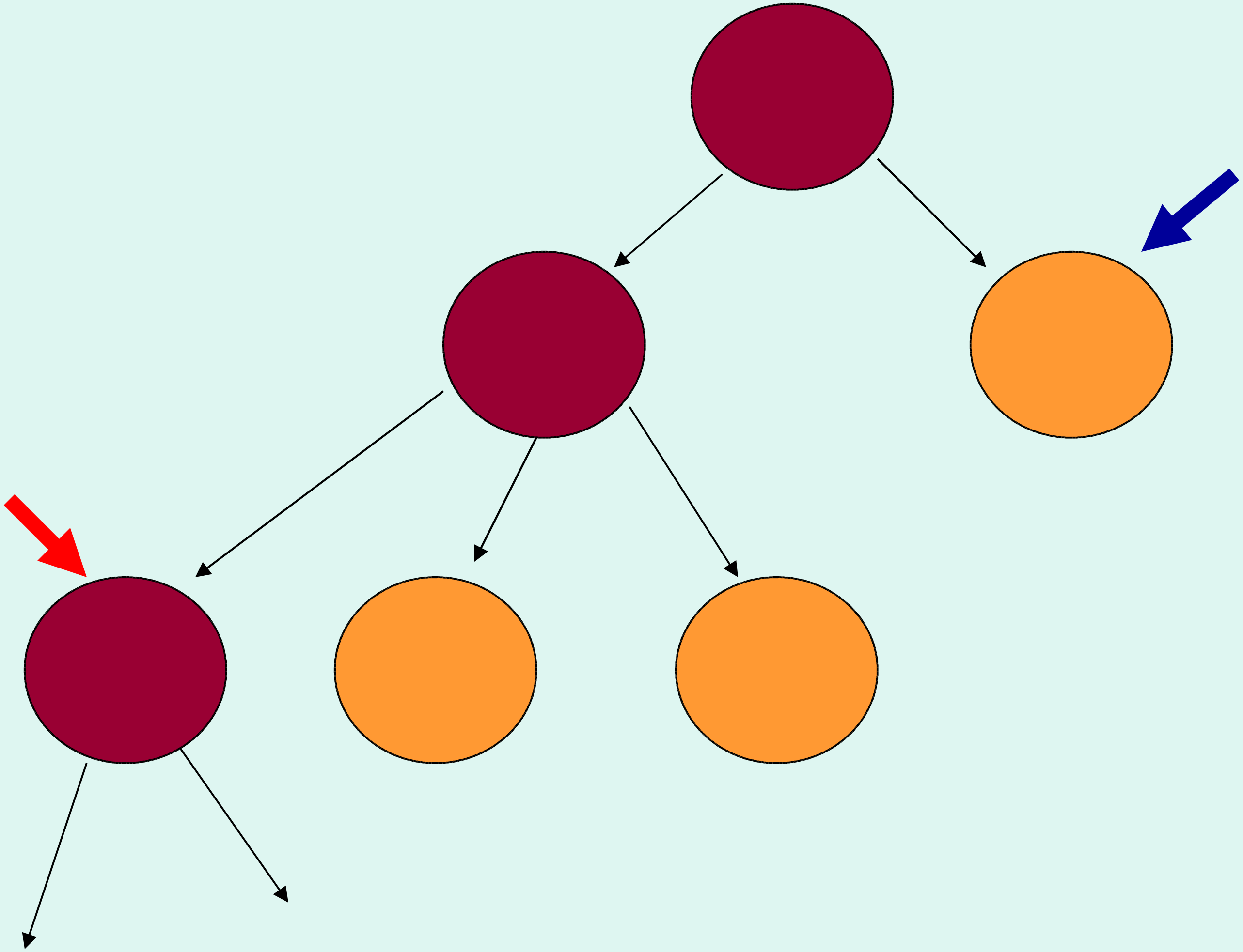
- 🟧 On the list of pages to visit and cache
- 🟥 Cached Page
- ⬜ Linked Page

Web Crawler Example



- 🟧 •On the list of pages to visit and cache
- 🟥 •Cached Page
- ⬜ •Linked Page

Web Crawler Example



- 🟧 •On the list of pages to visit and cache
- 🟥 •Cached Page
- ⬜ •Linked Page

How Pages are Searched

- Need an efficient, fast way to search web pages
- Crawlers use a combination of policies:
  - *selection policy: which pages to download*.
  - *re-visit policy* : when to check for changes to pages
  - *politeness policy*: to prevent overloading websites
  - *parallelization policy:* how to coordinate multiple crawlers.
- Trade secret!
  - Search engines are big business and how they search the Web is not shared

http://en.wikipedia.org/wiki/Web_crawler

Standard for Web Pages

- Web pages must all be written in the same “language”
  - Browsers have to know how to display a web page
  - Should be universal
- This standard is HTML, or HyperText Markup Language

Describing a Document

- WYSIWYG
  - What you see is what you get
  - Examples: FrontPage, Word, Word Perfect
- Very commonly used
  - Easier to learn
  - Immediate feedback to the user

Markup

- Requires the use of a “markup language”
  - Much like a programming language
  - Uses special instructions, marks, or annotations to determine the appearance of the document
- Editing is commonly done with a simple text editor
  - Easier with special text editors that “color code” the different parts of the document
    - SuperEdi from the Course Software section of the website is an example
- Requires the document to be processed, then it can be see the finished version using a special *viewer*
  - Web browser is an example of a viewer

Why Use Markup?

- Separates *content* from *appearance*
  - Changing the appearance is often easier than with using a WYSIWYG
  - Allows for complex formatting of documents
- Does not usually require expensive editors or software
- Examples include: HTML, XML, LaTex

Types of Markup

- Physical / Visual
  - Describes how part of the document should look
    - Example: bold, italic, font size
- Logical / Structural
  - Describes the text according to its meaning or purpose
    - Example: title, paragraph, table caption, …
  - Appearance is determined by defining a *style*
    - So changing how your document looks is as simple as changing the style

Benefit of Logical Markup

- Change all chapter headings to 14pt, Times New Roman, with the Color Red
  - Probably requires changing at most a line or two with logical markup
  - With physical markup, have to find and change every chapter heading individually
- Can have different styles for different situations
  - Different sizes for visually impaired
  - A printed style

Markup and HTML

- HTML uses both physical and logical markup
- Logical markup can be used by some browsers to categorize your page correctly
- Can be used by other programs, such as text-to-speech programs

HTML

- Describes hypertext pages, more specifically, web pages
- Many different versions of HTML
  - We will be using XHTML 1.0
- XHTML: eXtensible HyperText Markup Language
  - For the most part, we will be using the terms HTML and XHTML interchangeable

XHTML

- XHTML 1.0 is the new web standard
- W3C: World Wide Web Consortium
  - http://www.w3.org/
- They aim to create a single web standard and guidelines for the web
- Why bother?
  - So there aren’t problems like with Internet Explorer, or even Opera

Basics of HTML

- Markup of an HTML document is done using tags
- Tags consist of a word, or even just a few letters as an abbreviation surrounded by triangular braces <>
  - <b> bold text
  - <p> paragraph
  - <h1> level 1 heading (the biggest)

Tags, cont.

- Tags have both an opening and closing version
- <b> is the opening tag for bold
- </b>is the closing tag for bold
- Opening and closing tags are identical, except the closing tag has a forward slash / preceding the content of the braces

HTML example

- Text:

  this is <b>bold text </b>

- Displayed as:

  this is **bold text**

- Content of the tag is the text between the opening and closing tags
- Content of the <b> tag is “bold text”

Common HTML Tags

- <h1> … </h1> level one heading
  - <h2> … </h2> level two heading
    - <h6> … </h6> level six heading
- <p> … </p> paragraph
- <i> … </i> italic
- <em> … </em> emphasized text
- <html> … </html> HTML document
- <header> … </header> header information
- <title> … </title> title (within header)
- <body> … </body> body of an HTML file

- XHTML reference in the references section of the course website

Closing Tags

- In older versions of HTML you didn’t always have to have closing tags
- XHTML 1.0 **requires** closing tags
- Some special cases where certain tags don’t have contents
  - <br> creates a line break
  - <br></br> is redundant
  - XHTML shorthand: <br />
    - Indicates that you want the tag closed immediately

Nesting Tags

- What if you want more than one tag for a certain portion of the text?
- You can *nest* tags
- Have to be nested properly
- Incorrect:
  - <b><i>This is bold and italicized</b></i>
- Correct:
  - <b><i>This is bold and italicized</i></b>

Restrictions on Nested Tags

- Some tags can only occur inside of other tags
- <p>, the paragraph tag, can only occur within the <body> tag
- <li>, list item tag, can only occur within the <ol>, ordered list, or <ul>, unordered list tags

Nesting Example

<ol><li>HTML is good</li><li>XHTML is better</li></ol>

Would be displayed as:

1. HTML is good
2. XHTML is better

Important Points to Remember

- All tags in XHTML **must** be lowercase
  - <BODY> and <Body> are wrong
  - <body> is correct
- If the browser doesn't know what to do with the tag, it ignores it
  - Contents of tag will still be displayed, but as plain text
- Different browsers will display HTML differently
  - Not WYSIWYG!
  - Properly used markup will allow your page to still be readable

Spaces and HTML

- Spacing within an HTML document is ignored
  - Hitting the space bar repeatedly
  - Enter or Return
  - Tab
- The following html text produce the same result
  - **I   love**
    peanut   butter and

    jelly
  - **I love peanut butter and jelly**
- All spacing must be accomplished using markup

Creating an HTML document

- In class example