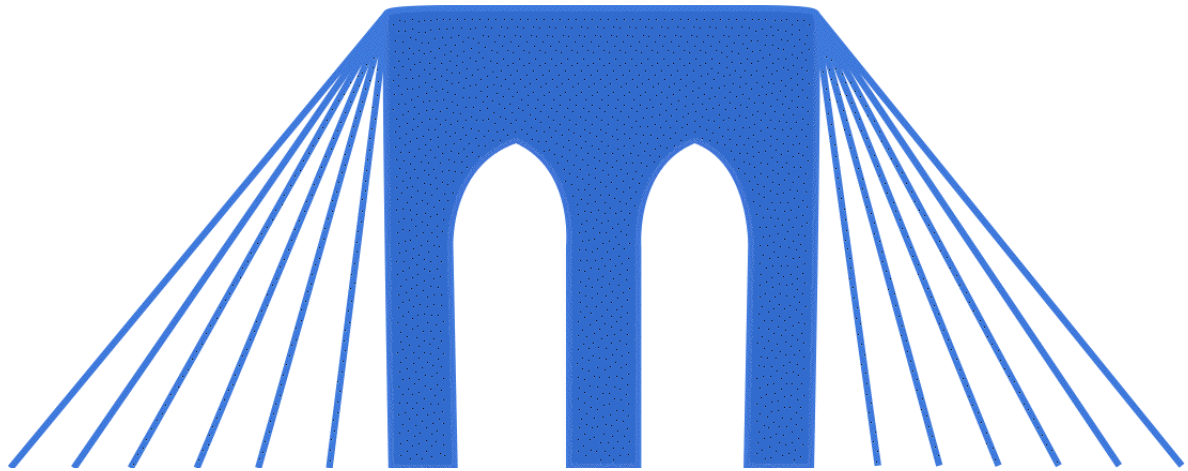
Bridges To Computing



General Information:

- This document was created for use in the "Bridges to Computing" project of Brooklyn College.
-
- For more information about the Bridges Program, please visit our website at: http://bridges.brooklyn.cuny.edu/

Disclaimers:

- **IMAGES**: All images in this presentation were created by our Bridges to Computing staff or were found online through open access media sites and are used under the Creative Commons Attribution-Share Alike 3.0 License. If you believe an image in this presentation is in fact copyrighted material, never intended for creative commons use, please contact us at http://bridges.brooklyn.cuny.edu/ so that we can remove it from this presentation.
- **LINKS**: This document may include links to sites and documents outside of the "Bridges to Computing" domain. The Bridges Program cannot be held responsible for the content of 3rd party sources and sites.

Introduction to Web Programming and Design

Lecture 1

Bridges to Computing
M. Meyer

Contents

Topics:

- Markup Languages
  - Definition
  - Motivation
  - Types
- XHTML
  - Overview
  - Rules, Validation
  - Examples
  - About Semantic Markup

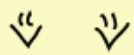
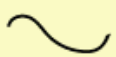
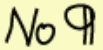
Question:

- What's a language?
  - Verbal languages
  - Written languages
  - Visual languages
  - Analog languages
  - Digital languages
  - Programming languages
  - Markup languages
- Simple Answer: "A medium for sharing/exchanging information".

Markup Languages (defined).

- Markup languages are not the same as programming languages.
  - Programming languages are used to create programs that control the behavior of a machine.
    - C/++/#, Java, Processing, PHP, Perl, Haskel, Prolog
  - Markup languages are used for adding information (sometimes called metadata) to text in a way which is distinguishable from that text.
    - GenCode, LaTeX, SGML, XML, HTML, XHTML
- It is possible to embed programming language statements/commands into a markup language.

Editing/Proofreading Symbols (a type of markup language)

Symbol	Meaning	Example
	**Insert a comma**	A stitch in time saves nine.
	**Apostrophe or single quotaion mark**	The woodchuck couldn't chuck wood.
	**Insert something**	An Apple a day keeps the doctor away.
	**Use double quotation**	"The better to see you with," said the wolf.
	**Use a period here**	Mary had a little lamb.
	**Delete**	The cow jumped over the ~~full~~ moon.
	**Transpose elements**	Jack over jumped the candlestick.
	**Close up this space**	The win dows on the bus goes up and down.
	**A space is needed here**	And they all lived happly everafter.
	**Begin new paragraph**	"Knock, knock." "Who's there?"
	**no paragraph**	"I'll huff and puff" said the wolf. "And I'll blow your house down!"

Motivation (WWW)

- It is possible to send pictures over the Internet, but pictures are large (slow to transfer), fixed (difficult to rearrange) and difficult to resize (interpolation, aliasing).
- It's very cheap and easy to send plain text over the Internet (ASCI, Unicode, UTF-8, UTF-16 and UTF-32). However plain text is difficult to read.
- **Markup languages allow us to add information to text, in a manner that is distinguishable from the text.**
- Markup languages can be used to enhance the comprehension/understandability of the text.

Text vs. Images

This very simple webpage illustrates why we use markup languages.

The letter A. The image

To see the difference, zoom in with your browser!
Letter A: 1 byte
Image A: 450 bytes

Plain Text (difficult to read)

COURSE DESCRIPTION. This course will commence with a broad introduction to topics in Multi-Media Computing (MMC), including: web design, game design, animation, data visualization, simulation and robotics. Discussions will be introductory and will cover a broad range of subjects, such as: multimedia hardware and software, including game boxes; human interface design and input using multi-media devices; graphical and other forms of output to multi-media devices; computer-based sound editing; agent-based programming for simulations and robotics; and uses of multi-media in industry. Emphasis is on the design and creation of a range of artifacts, including: web pages, with HTML and cascading style sheets; interactive, graphical web-based programs; and simple computer games and narratives. The format consists of alternative lecture/laboratory class sessions, with strong emphasis on hands-on learning. The following topics will be covered in 4 curricular units:Introduction to web programming and web design.(Princples of Web Design; XTHML; Cascading Style Sheets (CSS); GIMP; Audacity); Interactive web programming and data visualization; (Processing; Many Eyes; JavaScript; Server Side Scripting Languages); Game Programming and Visual Narratives; (Scratch; FLash CS4; Mobile Device Programming); Agent-based programming, simulations, multimedia devices; (NetLogo; PD).

Text augmented with presentation markup (easier to understand).

COURSE DESCRIPTION

This course will commence with a broad introduction to topics in Multi-Media Computing (MMC), including: web design, game design, animation, data visualization, simulation and robotics. Discussions will be introductory and will cover a broad range of subjects, such as: multimedia hardware and software, including game boxes; human interface design and input using multi-media devices; graphical and other forms of output to multi-media devices; computer-based sound editing; agent-based programming for simulations and robotics; and uses of multi-media in industry. Emphasis is on the design and creation of a range of artifacts, including: web pages, with HTML and cascading style sheets; interactive, graphical web-based programs; and simple computer games and narratives. The format consists of alternative lecture/laboratory class sessions, with strong emphasis on hands-on learning.

The following topics will be covered in 4 curricular units:

1. Introduction to web programming and web design.
   (Princples of Web Design; XTHML; Cascading Style Sheets (CSS); GIMP; Audacity)
2. Interactive web programming and data visualization
   (Processing; Many Eyes; JavaScript; Server Side Scripting Languages)
3. Game Programming and Visual Narratives
   (Scratch; FLash CS4; Mobile Device Programming)
4. Agent-based programming, simulations, multimedia devices
   (NetLogo; PD)

Types of Markup

1. **Presentational markup:** Used by traditional word-processing systems, to create a WYSIWYG effect.
   Examples: add line break, bold word, change font style or color.
2. **Procedural markup:** Provides instructions for programs that are to process the text.
   Examples: add an image, applet or link to a document.
3. **Semantic markup:** Used to label parts of document and attach additional meaning to those sections.
   Examples: define the title of a document or declaring that a section of text is an address.

Markup Languages - Key Terminology

- **Tag:** A markup construct that begins with "<" and ends with ">". Tags come in three flavors: start-tags, for example , end-tags, for example , and empty-element tags, for example
.
- **Element**: A component that begins with a start-tag and ends with a matching end-tag, or consists only of an empty-element tag. The characters between the start- and end-tags, if any, are the element's content, and may contain include other elements, which are called child elements. An example of an element is Hello, world. . Another is
- **Attribute:** A construct consisting of a name/value pair that exists within a start-tag or empty-element tag. In the example (below) the element img has two attributes, src and alt: .

Intro to XHTML

1. XHTML is an extension of HTML and stands for eXtensible Hyper-Text Mark-up Language.
2. XHTML is a language web servers can use to communicate with computers via web browsers.
3. XHTML content is delivered in pages, consisting of plain text interspersed with tags.
4. Web pages are stored as files on computers called servers, because they “serve” (i.e., deliver) the content to the computers that want to look at it.
5. Web content pages, “documents”, are stored in files with names ending in: .html or .htm

HTML vs. XHTML

- XHTML is almost identical to HTML 4.01
- XHTML is a stricter and cleaner version of HTML
  - Many pages on the internet contain "bad" HTML.
  - Browsers are still expected to interpret this "bad" HTML correctly (mobile devices).
- XHTML is HTML defined as an XML application
  - This allows you to create and define your own tags.
- XHTML is a W3C Recommendation and is designed to completely replace HTML 4.
- NOTE: HTML 5 is NOT a W3C recommendation, and no browser has FULL HTML 5 support... yet.

XHTML Rules (1)

The Most Important Differences ARE:
(all examples are WRONG)

1. XHTML elements must be properly nested
   - EX: This text is bold and italic
2. XHTML elements must always be closed
   - EX: <p>This is a paragraph
3. XHTML elements must be in lowercase
   - EX: <P>This is a paragraph</P>
4. XHTML documents must have one root element
   - EX:

```
<html>
    <head> ... </head>
</html> <html>
    <body> ... </body>
</html>
```

XHTML Rules (2)

Some More XHTML Syntax Rules

(all examples are CORRECT)

5. Attribute names must be in lower case
   Ex: <table width="100%">
6. Attribute values must be quoted
   Ex: <table width="100%">
7. Attribute minimization is forbidden
   Ex: <frame noresize="noresize" />
8. The id attribute replaces the name attribute
9. XHTML has predefined mandatory elements

Minimum Components of a "Transitional" XML document

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

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

      <head>
            <title>simple document</title>
      </head>

      <body>
            … some content …
      </body>

</html>
```

Document Type Definitions (DTD) [XML-Schema]

- A DTD specifies the syntax of a web page in SGML. DTDs are used by SGML applications, such as HTML, to specify rules for documents of a particular type, including a set of elements and entity declarations.
- An XHTML DTD describes the precise, allowed syntax of XHTML markup
- There are three XHTML DTDs:
  - TRANSITIONAL
  - STRICT
  - FRAMESET

XML Documents Can Be "Validated"

- You can check your .html documents to see if the are "valid" by going to the following link: http://validator.w3.org/
- If your xhtml file violates any rules or is missing any required elements it will generate errors. The "minimal transitional document" from the previous slide is free from errors (although it generates "warnings").
- XHTML pages (and websites) that are "valid" can then add the following picture:

Writing XHTML

- Many applications can be used to create XHTML documents. No matter what you use, the basic underlying XHTML is the same.
- For this class, you will use a text editor to write basic XHTML:
  - on a PC, this is Notepad (not Wordpad) or Notepad++
  - on a Mac, this is TextEdit (in plain text mode, not rtf)
  - on Linux/Unix, this can be pico or emacs or vi or Text Editor.
- For this class, you will create a file using a text editor and type content and XHTML tags into that file. Follow the lab sheet for detailed instructions!
- In the remainder of these notes, you will find a quick reference to basic XHTML tags.

If time is short…
proceed to lab 1

Page Tags

Doctype declaration has to appear at the top!
All other elements must be between the HTML tags.

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

<html>
   <head>
      <title>Page Title (Appears in browser box)</title>
   </head>
   <body>
      <h1>An important heading</h1>
      <h2>A slightly less important heading</h2>
      There are other sizes, from <h1> (largest) to <h7> (smallest)
      <p>A simple paragraph</p>
   </body>
     <!-- Comment goes here -->
</html>
```

Note the difference between header and heading: header appears at the top of the page, between <head>... </head> tags; whereas headings appear in the content of a page

Paragraph Tags

Paragraphs:

This is the first paragraph.
This is the second paragraph.

Line breaks:

This is the first line.
This is the second line>.

Division

A specific section of text

Note: The div element is very often used with CSS to layout a web page. Browsers usually place a line break before and after the div element.

Character Tags

Emphasis: This interesting!
Italics: This is really interesting!
Bold: This is a boldly interesting!
Preformatted Text:

```
<pre>
Text in a pre element
is displayed in a fixed-width
font, and it preserves
both      spaces and
line breaks
</pre>
```

Link Tags

A link is a tag that directs your browser to another page if the user clicks on the link. The content of the link tag is:

1. The URL where you want the browser to go if the user clicks on the link, and
2. The text that you want the user to see (i.e., the text that the user will click on to activate the link)

Examples:

This a link to
google.
This a link to
<a href="http://www.google.com"
Target=_blank>google
that opens up another window.

List Tags

Un-ordered list:

```
<ul>
    <li>the first list item</li>
    <li>the second list item</li>
    <li>the third list item</li>
</ul>
```

Ordered list:

```
<ol>
    <li>the first list item</li>
    <li>the second list item</li>
    <li>the third list item</li>
</ol>
```

Note: You can nest lists!

Table Tags

- Begin a table with <table>
- End a table with </table>
- Begin each row with <tr> and end each row with </tr>
- Begin each column with <td> and end each column with </td>
- Options:
  - borders
  - cellpadding (padding within a cell)
  - cellspacing (spacing between cells)
  - width and height (in pixels)
- Tricks:
  - empty cells — use
  - spanning multiple rows or columns
- Coloring cells:
  - <td bgcolor="red">ASDF</td>

Table Tags (cont)

- Aligning cell content:
  - horizontally: left, center, right
  - vertically: top, middle, bottom
- Advanced formatting:
  - cell padding (extra space inside the cells)
  - cell spacing (space between the cells)
  - width of table, of cells
  - alignment of cell content
  - empty cells ( )
  - multi-row and multi-column cells
  - borders
  - coloring cells
  - headings

Fonts & Deprecated Tags

We will talk about fonts when we cover CSS (cascading style sheets) the use of the font **tag** is deprecated in XHTML.

Other deprecated **tags**:

<xmp>
<u>
<strike>
<s>
<menu>
<isindex>
<dir>

<basefont>
<applet>

Semantic Markup

- In XHTML you can create your own tags.
- Why would you do this? To add meaning and clarity to pieces of information.
- These additional tags may also come from a predefined onotology (FOAF, DublinCore).

Example1:

```
<address>
   <name>Brooklyn College</name><br/>
       <number>2900</number><street> Bedford Avenue</street><br/>
       <city>Brooklyn</city>, <state>New York</state>
    <zip>11210<zip><br/>
   <phone>718.951.5000</phone>
</address>
```

Semantic Markup (cont)

Example2:

```
<acronym title="North Atlantic Treaty Organization">
NATO </acronym>
```

Example3:

```
<div class="column" id="column-Web20-design">
    <span class="title">Web 2.0 Design: Bootstrapping
    the Social Web</span> by
    <span class="author">Porter, Joshua</span> and
    <span class="author">MacManus, Richard</span>
    </div>
```

Ontologies

An ontology is a type of online dictionary/thesaurus that is usable by a computer program.

```
<?xml version="1.0" encoding="UTF-8"?>
	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
	"http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
	<html	xmlns="http://www.w3.org/1999/xhtml"
		xmlns:foaf="http://xmlns.com/foaf/0.1/"
		xmlns:dc="http://purl.org/dc/elements/1.1/"
		version="XHTML+RDFa 1.0" xml:lang="en">
		<head>
			<title>John's Home Page</title>
			<base href="http://example.org/john-d/" />
			<meta property="dc:creator" content="Jonathan Doe" />
		</head>
		<body>
			<h1>John's Home Page</h1>
			<p>My name is <span property="foaf:nick">John D</span> and I like
			<a href="http://www.neubauten.org/" rel="foaf:interest" xml:lang="de">
		Einstürzende Neubauten
		</a>.
			</p>
			<p> My <span rel="foaf:interest" resource="urn:ISBN:0752820907">
		favorite book</span> is the inspiring <span about="urn:ISBN:0752820907">
		<cite property="dc:title"> Weaving the Web </cite> by
			<span property="dc:creator"> Tim Berners-Lee </span>
		</span>
			</p>
		</body>
	</html>
```

Ontologies Example:

Where to get help

XHTML:

http://www.w3schools.com/xhtml/

HTML & XHTML Tags:

http://www.w3schools.com/tags/

The End