CISC 1600
Introduction to Multi-media Computing

Summer Session II 2012
Instructor : J. Raphael

Email Address: raphael@sci.brooklyn.cuny.edu
Course Page: http://www.sci.brooklyn.cuny.edu/~raphael/cisc1600.html
Class Hours: MTWTH 11:50 – 1:35PM 5301N

CISC1600-SummerII2012-Raphael-lec2

Agenda

• Markup Languages

• XHTML

• Tags

CISC1600-SummerII2012-Raphael-lec2

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.

CISC1600-SummerII2012-Raphael-lec2

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.

CISC1600-SummerII2012-Raphael-lec2

Text vs. Images

This <u>very simple webpage</u> illustrates why we use markup languages.

> The letter A. The image **A**
>
> To see the difference, zoom in with your browser!
> Letter A: 1 byte
> Image A: 450 bytes

CISC1600-SummerII2012-Raphael-lec2

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).

CISC1600-SummerII2012-Raphael-lec2

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)

CISC1600-SummerII2012-Raphael-lec2

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.

CISC1600-SummerII2012-Raphael-lec2

Markup Languages - Key Terminology

- **<u>Tag:</u>** A markup construct that begins with "<" and ends with ">". Tags come in three flavors: start-tags, for example <p>, end-tags, for example </p>, and empty-element tags, for example <br/>.
- **<u>Element</u>**: 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 <p>Hello, world.</p>. Another is
- **<u>Attribute:</u>** 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: .

CISC1600-SummerII2012-Raphael-lec2

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

CISC1600-SummerII2012-Raphael-lec2

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.

CISC1600-SummerII2012-Raphael-lec2

XHTML Rules (1)

<u>The Most Important Differences ARE:</u>
(all examples are WRONG)

1. XHTML elements must be properly nested
   - EX: <i>This text is bold and italic</i>
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>
     ```

CISC1600-SummerII2012-Raphael-lec2

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

CISC1600-SummerII2012-Raphael-lec2

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>
```

CISC1600-SummerII2012-Raphael-lec2

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

CISC1600-SummerII2012-Raphael-lec2

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: W3C XHTML 1.0

CISC1600-SummerII2012-Raphael-lec2

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.

CISC1600-SummerII2012-Raphael-lec2

Page Tags

•opening tag: <html>

•closing tag: </html>

•header:
      <head>... </head>

• body:
      <body>... </body>

• title (goes inside the header):

  •<title>My first HTML document</title>

  •this text appears as the title of the browser window

CISC1600-SummerII2012-Raphael-lec2

Page Tags

•heading

  •<h1>An important heading</h1>

  •<h2>A slightly less important heading</h2>

  •there are other sizes, from <h1> (largest) to <h7> (smallest)

• 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

CISC1600-SummerII2012-Raphael-lec2

Paragraph Tags

•paragraphs:

      <p>This is the first paragraph.</p>
      <p>This is the second paragraph.</p>

•line breaks:

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

CISC1600-SummerII2012-Raphael-lec2

Character Tags

•emphasis:
   This is a really interesting topic!

• italics:
   This is a really <i>interesting</i> topic!

•bold:
   This is a boldly interesting topic!

•preformatted text:
   <pre>
      hello world
      hello
      again
   </pre>

CISC1600-SummerII2012-Raphael-lec2

Font Tags

•font size:
   <big>this text is very big</big>
   <small>this text is very small</small>

•another way to set font size:
   this text is in font size 1 (smallest)
   this text is in font size 7 (largest)

• font face:
   this text is in courier font face

• font color:
   this text is red

CISC1600-SummerII2012-Raphael-lec2

Links

•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 consists of:

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

•example:
   This a link to google.

   This a link to google that opens up another window.

CISC1600-SummerII2012-Raphael-lec2

Lists

•Un-ordered list:

      the first list item
      the second list item
      the third list item

•Ordered list:

      the first list item
      the second list item
      the third list item

CISC1600-SummerII2012-Raphael-lec2

Nested Lists

•Lists can be nested, which means you can put one list inside another list:

•this is my shopping list:
```
<ul>
    <li>bread</li>
    <li>apples</li>
    <li>ice cream</li>
    <ul>
        <li>vanilla</li>
        <li>chocolate</li>
        <li>mango</li>
    </ul>
    <li>peanut butter</li>
</ul>
```

CISC1600-SummerII2012-Raphael-lec2

Tables

•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)

CISC1600-SummerII2012-Raphael-lec2

Tables Continued

•tricks:
- empty cells --- use
- *spanning* multiple rows or columns

•coloring cells:
```
<td bgcolor="red">ASDF</td>
```

•aligning cell content:
- horizontally: left, center, right
- vertically: top, middle, bottom

CISC1600-SummerII2012-Raphael-lec2

Tables Continued

•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

CISC1600-SummerII2012-Raphael-lec2

Tables Continued

```
<table>
    <tr> <td>Name</td>  <td>Score</td></tr>

    <tr><td>Apple</td><td>9</td></tr>

    <tr><td>Bob</td><td>12</td></tr>

    <tr><td>Carl</td><td>3</td></tr>
</table>
```

CISC1600-SummerII2012-Raphael-lec2