HTML

Web Building Basics for RPG programmers

Jerome Hughes

OMNI Technical Conference 2003

HTML: what is it?

- HyperText Markup Language
- Originally derived from SGML
    - Simple Generalized Markup Language
    - Used for document markup
        - Mostly technical, legal
- HTML created for Web

HTML history

- Invented at CERN European Particle Physics Lab
- Browser implemented at U of I Mosaic
- Standards from W3C
    - World wide web consortium - w3.org
- Varied support of standards
- Best to use what works for most

Client requests document

- Uses HTTP   hypertext transfer protocol
- Names equated to IP by DNS
- http://   hypertext transfer protocol
- www.   server name
- omniuser   domain name
- .org   high level domain - .com, etc.

Server returns document

- Content
  - text and images about subject
- Markup Instructions
  - tags specify structure and formatting
  - simplest is HTML, many other flavors

Browser renders document

- According to standards
- Watch out for browser differences

Compose and test offline

- Create documents in most any editor
- Open… File… in browser
- Iterative development cycle
    - Edit and save document
    - Open or refresh in browser
    - and so on…

Root of related technologies

- XML - extensible markup language
- CGI - web server extensions
- .asp, .jsp, .php - server-side scripting
    - Insert db info in template page with scripting evaluated at retrieval time
- For any of these, need a good understanding of how HTML works

Start with static pages

- can generate/convert with many tools…
  - Word, Excel, Access… export to .html
- Frontpage, Dreamweaver… “CASE” tools
- Useful, but no replacement for understanding how code constructs work
- To generate or “script” first understand using constructs “raw” or by hand

HTML as XML: XHTML

- XML is eXtensible Markup Language
- Like SGML, XML is a meta-language
  - a language for expressing languages
- XHTML is HTML implemented in XML
- Stricter syntax, more machine verifiable

Extensions to Standards

- Browser compatibility issues
- Extensions for competitive advantage
- Often adopted eventually
- Know your audience
- Lowest common denominator or multiple choices for different usages

Tools inventory

- Text or WYSIWYG editor
  - Either way, become fluent in "native" HTML
  - WYSIWYG "generated code" can be annoying
  - If you understand, you can leverage
- Browsers
  - Internet Explorer, Netscape Navigator
  - Other browsers (many) exist, but not as important
  - Must adhere to IE and Netscape standards

Hello, World (Wide Web)

```
<html>
<head>
<title>My HTML document</title>
</head>
<body>
<h2>My HTML document</h2>
Hello, <b>World Wide Web!</b>
<p>
Go to <a href="http://omniuser.org>omniuser.org</a>
<p>
&copy; 2002 omniuser.org
</body>
</html>
```

Save .html, open in browser

- Browser renders file, showing text, reacting to markup (tags)
- Tags are text bracketed between less than (<) and greater than (>) symbols
- Tags are embedded into text document opened and viewed with browser

Tags start and end

- <tag> opens a "section" of code, continues until matching </tag>
- For some tags, </tag> end tag is optional
- Opening <tag> can specify attributes
    - <tag option="value">
- Closing tag doesn't specify attributes, closes effect of opening tag

<head> and <body> tags

- Two parts of document
- Both inside <html> … </html> “container”
- Both are containers as well
- <head> = information about document
    - important: <title> displayed as window title
- <body> = document content
    - where most work occurs

Comments

- Opening comment tag is <!--
- Nothing after it is processed until
- Closing comment tag is -->
- Allows explanations
- Also useful to “turn things off”

Text

- What's between the tags
- What the user sees
- It's the content that matters most!

“Anchor” tags for links

- To other documents
    - <a href=http://omniuser.org>Omni home page</a>
- To images
    - <img src=“images/photo.jpg”></img>
    - <img src=“images/photo.jpg”/>

A link that's also an image

- <a href="http://omniuser.org><img src="images/omniImage.jpg/></a>
- Containers hold other elements
- Formatting tags (deprecated)
  - <i> ... </i>, <b> ... </b>
- Character entities
  - &copy; &nbsp; &gt; &lt; &amp; (!)

Text processing is different

- White space is ignored
- Returns are ignored
- Use <div> <p> and <br> to force display
    - All break to newline
    - <div>, <p> define section, break line
    - <br> just causes a line break

Headings and rules

- Headings
  - <h1> through <h6>
  - Bolded, sized accordingly
- Rules
  - <hr>
  - Separation lines

Hyperlink sites & documents

- Also called "anchor" tags
- absolute
  - <a href="http://omniuser.org">Omni User</a>
  - without document, defaults index.html
- relative
  - <a href="press/article.html">Interesting!</a>
  - links to documents in your own site

Inserting images

- Inline images
    - <img src="images/photo.jpg"/>
    - <img src="images/simple.gif" />
    - Flowed with text like big characters
    - Vertically align with align attribute
        - <img align=top|middle|bottom src=.../>

Image maps

- In <a … > tag, use <img ismap … > attribute to define areas
- Browser returns x, y coordinates when image clicked
- Processed by server program

Lists

- Unordered lists
  - <ul><li>item 1</li><li>item 2</li></ul>
  - Indented with bullets
- Ordered lists
  - <ol><li>item 1</li><li>item 2</li></ol>
  - Indented with numbers

More lists...

- Definition lists
    - <dl><dt>title1</dt><dd>defn1</dd>
    - <dt>title2</dt><dd>defn2</dd></dl>
    - List of titles and indented related definitions

Forms handle user input

- <form> input elements </form>
- Fields, text boxes, radio buttons, check boxes (more on these later...)
- Submit button sends information to be processed by server

Tables divide page space

- <table>...table specs...</table>
    - within, one or more <tr>...row specs...</tr>
    - within, one or more <td>...data specs...</td>
    - Span attribute allows bridging cells
- <table>
    - <tr><td>r1c1</td><td>r1c2</td></tr>
    - <tr><td>r2c1</td><td>r2c2</td></tr>
- </table>

Frames

- Divide window, load multiple .html documents into divisions
- <body> replaced with one or more <frameset> tags
- Each <frameset> tag tells browser what document to load in that window space

Tag syntax

- All tags have a name, some have attributes
- HTML is case insensitive, XHTML is all lowercase, so use lowercase for future
- <tagname {att1="val1" {att2="val2"}}>
- Order of attributes does not matter
- XHTML requires value quotes, so use 'em

Nesting tags

- End embedded tag before ending enclosing tag
    - this <a href=“http://omniuser.org><b>omni</b></a>
    - not <a href=“http://omniuser.org><b>omni</a></b>

Some tags have no end tag

- In HTML, especially <br>, <hr>, <img>
- XHTML requires end tags, well formed
- Also, some tags are ignored
  - Redundant tags <p><p> vs. <br><br>
  - Especially tags incorrectly specified

Combine and investigate

- Look at examples on the web
- Or Save As… .html from Word
- Display in browser, then use View.. Source
- Copy text, modify, redisplay
- Replace content with your content

Colors and backgrounds

- <body bgcolor="#FF0000">
- <body bgcolor="black">
- <body background="images/backg.gif">
    - Backgrounds tile or repeat if they aren't big enough, small graphics create interesting effects when tiled
- <body text="white">
    - change text color to contrast

Link colors

- link, vlink, alink color attributes
  - link, visited link, active link
- but should such things be specified for each link?

CSS externalizes formatting

- Cascading Style Sheets separate content and format
- <link rel="stylesheet" type="text/css" href="/std.css" />
- “This sort of tag, in this situation, should look like this”
- selector {property: values; property: values;... }

  -
    ```
    p,table,li,h1,h2,h3
    {
    font-family: verdana, arial, 'sans serif';
    }
    ```

Cascading?

- Many styles can apply to an element
- Browser checks, first to last...
    - Inline Style (inside HTML element)
    - Internal Style Sheet (inside the <head> tag)
    - External Style Sheet
    - Browser default

Styled links (internal sheet)

-
  ```
  <style type="text/css">
  a:link {color: #FF0000}
  a:visited {color: #00FF00}
  a:hover {color: #FF00FF}
  a:active {color: #0000FF}
  </style>
  ```
- good start point: add to head, remove html formatting
- when comfortable, externalize and reference

Source style concerns

- How it looks in document does not equal how it looks in browser
- In document, arrange for understandability and maintenance
- Can look same in browser, but be easier to live with
- Nesting is your friend

Forms for user input

- Can be many on a page
- A typical form
- <form action="http://omniuser.org/process">
    - … form elements, including text boxes, buttons, etc.
- </form>
- When submit button pressed, action document requested with current form element values passed along to server in request

Method - POST or GET

- <form> also requires method attr
- Values are POST and GET
- POST encodes form data in stream
  - Good for lots of fields, secure
- GET appends form data on URL
  - Good for few fields, direct links

Form example

- <form method=GET action="http://omniuser.org/process">
    - Name:
        - <input type=text name=name size=30 maxlength=80>
    - <p>Sex:
        - <input type=radio name=sex value="M" >
        - <input type=radio name=sex value="F">
    - <p><input type=submit>
- </form>

Email from forms

- <form method=POST action="mailto:jromeh@aol.com" enctype="text/plain" onSubmit="window.alert('This form is being sent by email…')">
- …
- </form>
- Sends form values as email, process transactions manually until automation required
- CGI better, but if security not paramount…

Text fields and file control

- <input type=text name=comment>
- <input type=text name=address size=30 maxlength=256>
- <input type=password name=pwd>
  - Only masked on screen, sent clear, be careful
- <input type=file size=20 name=myfile>
  - Displays file selection control

Checkboxes

- <input type=checkbox name=sys value="iSeries" />iSeries
- <input type=checkbox name=sys value="AS/400" />AS/400
- Allows multiple selections, sends as
  - sys=iSeries
  - sys=AS/400

Radio Buttons

- Like checkboxes, but only one in group may be selected
    - <input type=radio name=sys value="iSeries"/>iSeries
    - <input type=radio name=sys value="AS/400"/>AS/400
    - Returns sys=iSeries or sys=AS/400

Action buttons

- Send entire form when clicked
    - Submit button
        - <input type=submit>
        - <input type=submit value="Send It!">
    - Reset button
        - <input type=reset>
    - Image button
        - <input type=image src="images/btn.gif">
- Use value attribute to identify multiple submit buttons

Hidden fields

- Used to identify “author-time” constant values, like compile time constants
- <input type=hidden name=formnbr value=“002”>
- Passed through to server, not under user control

Text Areas

- <textarea name=address cols=40 rows=4>default data</textarea>
- Wrap attribute controls wrapping style
  - wrap=off - one line sent, displayed
  - wrap=virtual - one line sent, broken in display
  - Wrap=physical - two lines sent

Multiple choice elements

- <select name=lang size=3 multiple>
  - <option>RPG</option>
  - <option>COBOL</option>
  - <option>Java</option>
- </select>
- Shows 3 of 3, allows multiple selections

Multiple choice elements

- <select name=lang size=1>
    - <option selected>RPG</option>
    - <option>COBOL</option>
    - <option>Java</option>
- </select>
- Shows 1 of 3, starts with RPG selected, allows only one selection

JavaScript automates client

- Not Java at all!
- Treats document elements and browser as objects with properties
- Used in many tools (ECMA Script)
- Great for edits before submitting form

Explore & express content

- Clone some pages, adjust content and layout to your needs
- Pay attention to content and consistency
- Use web resources
- Get some server space and post pages
- Regular content change = regular visits!

Web resources

- Google
  - Search on html tutorial
  - NCSA, WebMonkey, w3.org, w3schools.org and on and on…
- Have fun!

HTML

Web Building Basics for RPG programmers

Jerome Hughes

OMNI Technical Conference 2003