

CSC 551: Web Programming

Fall 2001

emerging & alternate Web technologies

- Dynamic HTML
- ActiveX
- XML

course overview

- online review sheet
- advice for the final exam

Dynamic HTML

chapters 30 & 31 describe more dynamic features of HTML

- possible to change the appearance of HTML elements after loading
- Netscape utilizes JavaScript Accessible Style Sheets (JASS)
can reassign element attributes with JavaScript assignments
e.g., `document.tags.h1.fontSize="18pt";`

each browser provides a different Scripting Object Model

- provide mechanisms for accessing/changing element attributes
- also, can handle events that happen in the page
- Microsoft example: `document.all.tags("h1").style.fontSize="18pt";`
- can define general event-handlers
e.g., when mouseover a menu, display alternative choices

ActiveX

ActiveX is a set of technologies from Microsoft

- provides tools for linking (Windows) desktop applications to the Web.
- ActiveX controls are the building block of applications
 - e.g., text control to read user ID & password, button control to submit
- similar to applets, but have full access to Windows OS
- once downloaded to the client, the control automatically registers itself & becomes available to the browser
- can automatically trigger a self-update if newer version is available

thousands of ActiveX controls are available for download

- can develop your own using Microsoft programming tools
 - e.g., Visual Basic, Visual C++, Visual J++

ActiveX controls are integrated into Microsoft products

- e.g., can allow users to view Word and Excel documents directly in a browser

Extensible Markup Language (XML)

protocol for representing structured data in text files

- can represent arbitrary structures, define own abstractions
- since raw text, easy to peruse/edit & platform-independent

XML is a meta-language for designing your own markup language

- like HTML, utilizes tags and attributes (e.g., `<p name="foo">`)
 - however, HTML specifies what each tag & attribute means
 - whereas, XML simply delimits pieces of data, no interpretation implied

XML is meant to be read by applications, not people

- formatting rules are very strict (missing tag or quotes invalidates file)
- many applications have been developed that utilize XML as data format

note: representing data as XML text is not the most efficient bitwise

- disk space is cheap; compression tools can alleviate

XML & HTML

HTML has been reformulated in XML (now known as XHTML 1.0)

- an existing HTML document is valid XML as long as
 - first line is of form: `<?XML version="1.0" rmd="none" ?>`
 - tags & attributes are lower-case, attribute values are in quotes
 - every opening tag has a closing tag (or else ends with />)
e.g., `<p></p>` ``

can define own tags for structuring data

```
<question>
  Where was the Web invented?
  <answer> Microsoft </answer>
  <answer> Sun Microsystems </answer>
  <answer correct="true"> CERN </answer>
  <answer> IBM </answer>
</question>
```

DTD's and XSL

Document Type Declaration (DTD) specifies the element structure

```
<!element question (#PCDATA, answer+)>
<!element answer (#PCDATA)>
<!attlist answer
  correct (true | false) "false">
```

- include the DTD with the XML document
automatically used to validate the element structure in the document

```
<?XML version="1.0" rmd="all">
<!DOCTYPE test SYSTEM "test.dtd"
<question>
...
</question>
```

Extensible Stylesheet Language (XSL)

- can define a stylesheet to tell the browser how to display the XML content

Next week...

final exam

- similar format to midterm
 - true or false
 - discussion/short answer
 - explain or modify code (HTML, JavaScript, Java, CGI)

- cumulative, but will emphasize material since the midterm

- designed to take 90-120 minutes, will allow full 165 minutes

- study hints:
 - review lecture notes
 - review text
 - look for supplementary materials where needed (e.g., Web search)
 - think big picture -- assimilate the material!
 - use online [review sheet](#) as a study guide, *but not exhaustive*