Presenting Basic Web Technology to Undergraduate Students through the Creation of Student Web Sites

Tom Wulf

College of Applied Science, University of Cincinnati

Abstract:

A one-quarter, undergraduate course in Web technology is described in which Information Technology students learn XHMTL, CSS, HTTP basics, Web site design, and implementation. Students re-enforce their knowledge of this material through a series of assignments in which they incrementally create personal Web sites. The Web sites created by the students persist after the course and this provides increased student motivation and interest in the material.

This course covers the basics of Web technology including the HTTP protocol, XHTML and CSS, ADA Guidelines and Web Design methods. There is no required text for this course[1]. Instructor lectures are presented in class and are available in MS power point format on the course's Blackboard (Bb) Web site for pre/re view along with a great deal of supplemental material. Some excellent Web-based tutorials on a 3rd party Web site[2] are assigned in lieu of textbook readings. Student assignments iteratively build on one another to cover the complete range of XHMTL and CSS syntax by requiring the student to create a personal Web page on the University server. This page persists after the completion of the course as long as the student remains at the University. I find that the creation of a lasting personal home page serves to motivate students with an enthusiasm that goes beyond what I normally see in my classes.

XHTML best practice is emphasized and students are required to test their XHTML code for compliance by using a validation utility as they complete each assignment.[3] They are also required to test their work for compliance with ADA design guidelines using the Bobbie[4] validation tool.

Students are taught the core set of (X)HTML structural markup tags. Browser-specific markup tags are deliberately avoided, since the goal is for the students to learn to create the most generic, platform-neutral Web documents.

Students are taught to use a simple text editor to create their XHTML documents rather than a Web development tool like MS Front Page or Dreamweaver MX, because these tools tend to prevent the learning of the XHTML syntax.[5]

Assignments are submitted by e-mailing the instructor and indicating that a given assignment has been completed. The student includes a hyperlink reference to his or her home page on the University server where the assignment Web document has been located. Thus, assignments are easily graded by inspecting the source code using a standard Web browser and students are permitted to correct and re-submit their work. Grades are e-mailed to the student along with any comments or feedback and are also recorded on the Bb course Web site.

Student accounts are created on a hosting server provided by the University and students are then taught the basic Unix file system commands they will need in order to create the content for the Web site. Generally, after the first few assignments, students tend to create their Web pages locally on a windows-based machines in the lab or at home and then upload them using the SSH file transfer tool. (Because of security issues with the TELNET and FTP applications, their use is discouraged in favor of SSH.)[6]

A description of the assignments follows:

Assignment #1: XHTML Basics, Online File Creation

XHTML syntax is introduced and students create an initial Web site consisting of a single XHTML document with a heading and a biographical paragraph about themselves. This assignment is completed online using the Unix pico editor although, a standard Unix editor, like vi or emacs could also be used[7]. Students are taught to use the SSH secure shell to connect to their

account on the host, and later are taught the SSH file transfer utility.

Assignment #2: XHTML Lists

In the next assignment, students create an example of each of the standard XHTML lists. They create a bulleted list consisting of the courses they are currently taking, an ordered list of their favorite movies, books, etc. and a definition list using whatever content items they wish.

Assignment #3: Hyperlinks, File Uploading, Use of Copyrighted Materials

In the next assignment, students create an additional Web document with a poem or song of their choice or a resume they have created and saved as a Web document using MS Word. The poem or song serves as a basis for a discussion about the use of copyrighted materials. Students are now taught to create Web documents with a local MS Windows editor, notepad and then transfer them to the Web server with the SSH file transfer utility. They hyper-link their new document to the original home page.

Assignment #4: Images, Mailto link and Internal Hyperlinks, ADA Guidelines

In the next, assignment, students add images, and both mailto and internal hyper links to the home page. The use of ADA guidelines is presented and the Bobbie ADA validation tool is introduced.

Assignment #5: Tables

In the next assignment, students add a separate new page to their Web site consisting of a variety of content including images, text, and lists of hyper links arranged within an XHTML table. This is also hyper-linked to the home page. Students are encouraged to develop this page around some theme of their own choosing. The use of XHTML tables for page layout, a pre-cursor to CSS is discussed.

Assignment #6: Frames

XHTML frames are now introduced. Students create a frame-based version of their Web site using the previous content from the home page and the separate pages already created. This page consists of a narrow, persistent title frame at the top of the page, a left-hand menu frame, and a larger right-hand frame that is used for a splash page and the display of the document content. (For instance, the 3 lists completed for the second assignment are divided into 3 separate pages that are displayed within the frames document.)

Assignment #7: CSS

CSS is now covered in the lecture sessions and students create a master CSS style sheet for their Web site and link all the separate pages to it. Additionally, students create a new sub page for their site that uses an embedded CSS style sheet which overrides the master sheet. This new sub page must also contain some of the more advanced features for page layout supported by CSS.

Additional Notes:

Currently, the material presented above represents the first half (five weeks) of a one-quarter (ten week) undergraduate course in Web technology. The second half of the quarter covers client-side Web programming with JavaScript. Again, programming assignments are implemented using the persistent student web sites.

In the past, I have taught this course with more of a focus on Web design and development. In that context, I used the last half of the quarter to cover Web design methodologies and the use of prevalent industry tools for Web site generation, instead of covering client-side programming. I added a project component in which the students develop a Web site for an external partner. Originally, this was a faculty member (course web site or home page), student group, or family business. Later I made the project into a Service Learning one that required the students to develop a Web site for a not-for-profit community partner. This required that they meet several

times with the community partner to complete the Web site design and then install the completed project on one of the free Internet hosting sites.[8]

Results:

I have been teaching Web technology since 1998. I've found that students respond very favorably to assignments that allow them to create persistent, personal Web sites. Because they have a personal interest in the work that they are doing for the course, they are generally more motivated to produce work of a higher quality rather than simply meeting the minimum requirements for an assignment as they typically do. In many cases, the students' enthusiasm leads them to go beyond the basic material that I present in the course. I've had students explore additional Web technologies including: the use of multimedia components, adding interactivity to a Web site by using canned Java applets, and creating password protected pages with restricted content, like family photo albums, among others.

I have also had a number of students who reported that they began doing Web development as a consulting business after completing this course. Typically, this starts with them completing a few Web sites for their friends and families using the various free hosting sites that are available and then they begin to pick up other clients via word of mouth advertising.

When I teach a more advanced Web programming course which focuses on server side programming using ASP .Net, I have to use a different server that supports the ASP technology. This server only hosts student-generated content for the duration of the quarter. I have noticed a distinct contrast in the attitude and the quality of the work submitted by students for this course in comparison to the work that they do on their personal Web sites for introductory course described here. When they are publishing their assignments to this temporary server, students tend to submit work that meets the minimal requirements for the assignments but rarely put any extra time in developing the look and feel of the sites they create or in experimenting with additional features and techniques.

¹ Although no required text is assigned for this course, optional texts are listed for those students whose learning styles work better with printed texts instead of the Web-based tutorials. (Generally, my experience is that there are usually one or two students from a typically class of twenty-four that prefer a textbook to the tutorials.)

² www.w3schools.com contains many excellent tutorials that cover basic Web technologies including: HTML, XHTML, CSS, JavaScript, XML, among others.

³ The W3 Consortium site at http://www.w3c.org/ contains a variety of validator programs for HTML, XHTML, CSS, and ADA Compliance. We also have these available locally on the University server.

⁴ ibid

⁵ A variation of this course for non-majors might include coverage of the prevalent industry Web generation tools instead of the client-side programming. I would favor Dreamweaver MX for this.

⁶ TELNET and FTP both transfer the account and password information as *cleartext* which is susceptible to a snooping attack. SSH encrypts all communications between client and server and is therefore preferred.

⁷ Generally, unless the students have already completed a basic UNIX course, I find that the simple pico text editor is a better choice than vi or emacs because it has almost no learning curve.

⁸ Yahoo Geocities, Tripod, etc.

TOM WULF
        Tom Wulf is an Assistant Professor of Information Engineering Technology at the College of Applied Science at the University of Cincinnati. He has a Masters Degree in Computer Science. His research and teaching interests include Web Technologies, Multimedia, IT Pedagogy, and Computational Linguistics with a focus on the development of web-based Computer Language Learning Systems for Old Norse.