

COURSE OUTLINE XHTML and CSS

Course Description

IN 112. XHTML and CSS. 3 hours credit. This course will enable the student to create web sites using HTML (Hyper Text Markup Language) and XHTML (Extensible HyperText Markup Language). In addition, the student will troubleshoot faulty web pages and provide corrective HTML coding. The student will learn about the origins of markup languages, how standards are established and changed, and the role of browser software companies in expanding web page capabilities. The student will hand-code XHTML and CSS using simple text editors. Before enrolling in the course, the student should be competent in basic computer operation.

Course Relevance

XHTML and CSS are foundational for a variety of web development professions, including web designer, multimedia programmer, interface designer, and more. Web developers use these markup languages to obtain more control over layout and functionality than is possible with web page editors. Knowledge of markup language coding makes it easier to write fast-loading web pages and to troubleshoot web pages. XHTML and CSS knowledge is foundational for learning more advanced Internet technologies such as scripting and forms development. It is also necessary preparation for advanced development tasks such as writing programs that output XHTML.

Required Materials

Freeman, E., and Freeman, E. (2006). *Head first HTML with CSS & XHTML* (1st ed.). Sebastopol, KS: O'Reilly.

USB Digital Drive: 1 GB minimum.

Note: Majors in digital media and web programming can use one drive for several classes

Learning Outcomes

The intention is for the student to be able to

1. Identify the entities responsible for implementing markup language standards
2. Code and troubleshoot HTML and XHTML web pages, incorporating CSS and scripts
3. Incorporate multimedia (images, animation, sound, and movies) into web pages using plugins and players
4. Identify and employ software and online tools used to troubleshoot markup language code

Primary Learning PACT Skills that will be DEVELOPED and/or documented in this course

Through the student involvement in this course, he/she will develop his/her ability in the following primary PACT skill areas:

1. Problem Solving
 - Through the analysis of technical issues related to markup language code, the student will demonstrate problem-solving skills.
2. Field-Related Technology
 - Through the selection and application of software and hardware employed to create and deliver web pages, the student will build markup language skills.

Secondary skills (developed but not documented):

Reading
Time Management
Internet Use
Computer Literacy

Major Summative Assessment Task(s)

These learning outcomes and primary Learning PACT skills will be demonstrated by

1. Preparing web pages that adhere to specifications of appropriate markup language organizations, demonstrate appropriate use of file management, and have been validated and quality-controlled

Course Content

- I. Themes – Key recurring concepts that run throughout this course:
 - A. Markup language tags, attributes and values interpreted by browser software
 - B. Technological evolution
 - C. File management
- II. Issues – Key areas of conflict that must be understood in order to achieve the intended outcome:
 - A. Balancing technical restraints and web page creation goals
 - B. Identifying constantly changing skills needed to produce markup language
- III. Concepts – Key concepts that must be understood to address the issues:
 - A. Evolution, distribution, and current implementation of markup languages
 - B. Relationship between markup languages and browser software
 - C. File management, paths, directories, and formats
 - D. Evolving markup tools and technologies
- IV. Skills/Competencies – Actions that are essential to achieve the course outcomes:
 - A. Identify the organizations involved in implementing markup language standards and credentialing professionals
 - B. Distinguish between the functionality of various markup language releases and browsers
 - C. Describe the relationship between HTML, scripting languages and programming languages
 - D. Manage files for effective web page development
 - E. Define the basic structure of XHTML tags, elements, attributes, and values
 - F. Using a simple text editor, compose markup code to create basic header, body, and meta tags

- G. Using a simple text editor, compose markup code to format text and add images to web pages
- H. Add internal, external and mailto links and image maps to web pages
- I. Create and apply styles to web pages
- J. Add lists, tables, frames and forms to web pages
- K. Describe deprecated (non-CSS) methods for controlling format and layout
- L. Implement CSS methods for controlling format and layout
- M. Troubleshoot faulty web pages
- N. Describe the process of publishing and publicizing web sites

Learning Units

- I. Identify the organizations involved in implementing markup language standards and credentialing professionals
 - A. The Internet before the visual web: DARPA
 - B. Tim Berners-Lee, CERN and Mosaic
 - C. The W3C (world wide web consortium)
 - D. CompTia and other agencies

- II. Distinguish between the functionality of various markup language releases and browsers
 - A. HTML 4.0
 - B. CSS 1.0
 - C. CSS 2.0
 - D. CSS 3.0

- III. Describe the relationship between HTML, scripting languages and programming languages
 - A. Markup languages
 - B. Scripting languages and macros
 - C. Procedural and object oriented programming languages

- IV. Manage files for effective web page development
 - A. Directory trees
 - B. Naming convention for Apache Linux and IIS servers
 - C. Moving down the directory tree
 - D. Moving up the directory tree

- V. Define the basic structure of XHTML tags, elements, attributes, and values
 - A. Browser interpretation of tags (HTML, XHTML and XML)
 - B. Symmetry in nesting elements; self closing tags
 - C. Assigning values to attributes; quotation marks and the difference between XHTML and HTML

- VI. Using a simple text editor, compose markup code to create basic head and body regions
 - A. Common contents of the head region

1. Title
 2. External CSS references
 3. JavaScripts, JScripts and external .JS references
 4. Meta Tags
 - B. Common contents of the body region
 1. Text
 2. Images
 3. Hyperlinks
- VII. Using a simple text editor, compose markup code to format text and add images to web pages
- A. CSS formatting; local, internal and external
 - B. Images: pathing and optimizing
- VIII. Add internal, external and mail to links to web pages
- A. Local hrefs: #
 - B. Local hrefs: server pages
 - C. External hrefs: http://
 - D. E-mail links: mailto:
- IX. Add lists, tables, frames and forms to web pages
- A. Ordered lists
 - B. Unordered lists
 - C. Tables, rows, columns and cells
 - D. Inline frames
 - E. Frames for FAQ pages
- X. Add scripts to web pages
- A. Hiding scripts from older browsers
 - B. Limits of server side scripting
 - C. JavaScript rollovers
 - D. JavaScript page redirects
- XI. Describe deprecated (non-CSS) methods for controlling format and layout
- A. Advantages of tabular layout
 - B. Disadvantages of tabular layout
 - C. Nested tables
- XII. Troubleshoot faulty web pages
- A. Validators
 - B. Symmetry in tabular layout
 - C. Differences in browser implementation of XHTML and CSS
 - D. Browser wars
- XIII. Describe the process of publishing and publicizing web sites
- A. FTP

- B. Domain names
- C. HTDOCS
- D. Index.html
- E. Search engine optimization
 - 1. Meta tags and crawlers/spiders/bots
 - 2. Reciprocal linkage
 - 3. Google bombing

Learning Activities

Learning activities will involve the student in examining various aspects of markup language, technology, and file management. Lectures, discussions, projects, readings, viewing of various pages, research assignments, quizzes and tests may be employed to prepare the student to be equipped to successfully complete the major assessment tasks.

Grade Determination

The student will be graded on assessment tasks, participation and the timely completion of class exercises and tutorials.