https://doi.org/10.48009/1_iis_2017_44-51

Issues in Information Systems
Volume 18, Issue 1, pp. 44-51, 2017

BACK TO TEACHING A WEB DESIGN COURSE AGAIN AFTER SOME BREAK

Azad Ali, Indiana University of Pennsylvania, azad.ali@iup.edu

ABSTRACT

This paper researches about the topics and teaching methods need to be included in the teaching of a web design course in light of the new and most recent technological advances. A faculty member has been teaching web design courses for years. This faculty took a break from teaching his web courses for a few years and came back to teach one of these courses again. In the few years break, web design technology has changed significantly. These changes have forced different faculty members to modify their courses to include teaching about the latest in technology. The author of this paper decided then to research about what he needs to include in his recent web design course so to accommodate the inclusion of teaching about latest in technology. The author intends to explain about his research findings and his experience about it in this in this study.

Keywords: Web design course content; Web design curriculum, Courses in web design, Mobile web design

INTRODUCTION

Information Technology (IT) faculty are too familiar with repeatedly updating the content of their courses. The updates are often required to cover changes in technology, modifications to teaching methods, and the adoption of textbooks that are more recent or for other reasons. More than any other field, web design courses face more repeated updates (Ali and Mensch, 2008, Bishop, Shuman & Vodnik, 2016, Craig, 2007). Numerous factors contribute to these rapid updates amongst, the multi technological interfaces with web pages (Fowler and Stanwick, 2004) and the different devices that used to display web pages (Bardzell, 2006). Understanding the interfaces with the web may require gaining knowledge about the technologies involved in the interface (Glassman and Shen, 2014). This may necessitate the coverage and explanation of these technologies in web design courses. Thus, changes in web design courses are too frequent and adopting the course to the most recent technological advances may be necessary.

This paper explains about the experience of a faculty teaching a web design course at a local university. This faculty taught different web design courses for years. However, scheduling constraints and departmental considerations did not permit the offering of web design courses for the past few years at the department. That has changed recently - the department has decided to offer a course in web design and the author of this study scheduled to teach it. The faculty author of this paper recognizes that a lot has changed in the web design field since he last taught it. He acknowledged the need to update the content and methodologies that he used in previous year in his teaching of web design courses. He also thought that the best way to learn about what is pertinent to teach in web design course is to conduct a study to investigate the more recent technological effects on the teaching of web design. Thus, this study investigates about the recent considerations for teaching web design courses. The author want to apply the findings in this study for his course in web design.

LITERATURE REVIEW – PARADIGMS OF WEB DESIGN COURSES

There are numerous paradigms in the information technology (IT) education that routinely contrasted when discussing the teaching of some IT courses. One example of a paradigm that has been discussed and for long is the teaching of OOP (Object Oriented Programming) first or teach it later (Ehlert, & Schulte 2009). Advocates of teaching it first note that teaching them from the beginning gets the students used to standards in technology and methodological development of programs from the start (Zhu & Zhou 2003). Meanwhile, critics of teaching OOP first note that including the teaching of OOP concepts from the beginning add another level of difficulty to an already difficult courses (Ben-Ari, 2010). The discussion continues about this topic and some educators elect to teach OOP first while others elect to teach it later – both sides have arguments to make in favor of their selection.

Similarly, the teaching of web design courses has its' own set of paradigms that are discussed and make faculty select one approach in the paradigm versus the other (Craig, 2007). Ali and Mensch (2008) explained about several paradigms within the context of teaching of web design. The list of paradigms discussed by Ali and Mensch was inclusive at the time. Nevertheless, more recent advances in the web technologies necessitated the introduction of other newer paradigms that are discussed when teaching web design courses. Thus, this section presents the paradigms that faced more recent changes lately. The remainder of this is section explains about the following contrasting paradigms in the teaching of web design courses:

- Mobile first and mobile later
- CSS first or CSS later
- Using code versus using software tools
- Free software versus propriety software
- Explorative learning versus learning by doing

Mobile First Versus Mobile Later

> Designing a website used to be simple. Users surfed the web on desktop computers. Though monitor sizes varied slightly, web sites designed to fit within a 1200-pixel-by-800-pixel box was the standard. Smartphones and tablets have forced web designers to think outside of that box. These days, web designers have no way of knowing what size screen might use to view a web site (Glassman and Shen, 2014, P. 78).

One of the main goals of designing web pages was (and still is) to design the pages in a way so that the content of the pages are correctly displayed on all devices (and all screen sizes). That is the web page content displayed without the need for excessive scrolling, screen resizing and moving around the displayed content of the web page. A secondary goal within this is to be able to show the same content legibly on all screen sizes. That was the goal from the beginning and no major issues raised about it at the time. Nevertheless, the latest introduction of various portable devices and mobile phones used to access the web has changed that.

At the beginning and through the many phases of teaching web design, one of the topics that were covered was the correct display of web pages on different screen sizes and different browsers. For most of the time, screen sizes were closely similar, so there was no major concern about dealing with this issue. Simple modifications of the web page may solve the problem and present the webpage in acceptable format on different screen sizes.

The introduction of different devices with much smaller screen sizes (like mobile phones and other portable devices) changed that a lot. Ganci and McCullough (2015) put the number of widely used mobile or tablet computers at 62 and noted that this number is subject to change by the time the article is released. Each of the screen sizes have other specific characters that may lead to displaying web content differently. So designers may need to account for these devices when they design their web pages (Gafni and Barak, 2013). In other words, the display of web pages on desktop is no longer a workable solution because it creates a lot of problems. As Glassman and Shen, 2014 articulated this point of problem "Viewing websites designed with desktop computers in mind on a mobile device can be frustrating, involving no small amount of scrolling, pinching, and screen rotation (p. 79)".

This increase of mobile phone use to access the web led to the creation of a new culture called by some the "mobile-first culture" (Kim, 2013). As a result, calls are repeated suggest the start of thinking of mobile users earlier as web sites are designed. Some suggest to consider mobile devices from the beginning as developers take steps to design web pages (MacConaonaigh, 2007). Another study suggests the need to design web sites for mobile devices first and then go back and adjust them for other devices (Smilansky, 2016).

These issues affected the teaching of web design as well. Teaching to design web sites for mobile devices is a must in these days. The question that is to ask at this point is whether teach it first from the beginning or whether to teach it later in the course after making some headway into the coverage of some of the course topics.

To teach it first, the students will get exposure to the more recent updates of technologies from the beginning. On the other hand, teaching it first makes it difficult from the beginning on the students to include explanation about the involved technologies. Natda (2013) specified these technologies as "HTML, CSS and JavaScript". To start teaching on desktop sizes and then moving down to include mobile devices gets the student used to learning on one size of technology which make it easier at the beginning. Moving to other devices later may sound seamless but it requires additional mental change of the learning.

CSS First Versus CSS it Later

To begin with, CSS stands for Cascading Style Sheets. Ernst (2002) explained the importance of CSS in web design "If you're not using CSS yet, you are missing a way to make web pages that are easier to create and maintain (P. 101)". So learning about CSS is essential to learning web design but the issue is whether to teach it from the beginning or wait until later and teach it.

A given web site contain two (or more) sets of documents: HTML documents and CSS documents. The HTML document contains the content of the web page (that is the text and the link to images and media files) while the CSS documents define the layout, the formatting and the presentation of the web page. Both documents are linked together to form the final display of the web page. As Blansit (2008) explained that the combination of these two documents separates the information on a web (saved in HTML file) from the stylistic instructions (saved in the CSS document).

HTML went through different revisions until it got to the current version that is now in the market which is HTML5. The earlier versions of HTML included sufficient coverage of formatting text and for different presentations. This enabled educators to go by teaching HTML for some time without the need for including discussion of CSS. Web design teachers had enough in HTML where they can format, align, decorate and present sufficiently without the need to using CSS. Nevertheless, that has changed, with the recent development of HTML versions. Newer versions of HTML, increasingly separates design from formatting and put the formatting tasks in the CSS documents. The latest HTML version (HTML5) separated this totally. Now, any formatting may need to be handled in the CSS document. So web design teachers cannot for very long go on without teaching about CSS (Motes, 2012).

CSS need be introduced and covered at earlier stages than before. But the question is whether to start teaching it from the beginning versus waiting some time before introducing it. Introducing CSS from the beginning teaches the students about standardizes formatting and presentations from the beginning, thus the students can practice standardized approaches from the beginning. Yet teaching it from the beginning adds an extra layer of difficulty of learning at a slower pace of progress because it involves the learning of other topics of the course. Students do not have to learn about tags only, they have to learn syntax of another language (though some does not consider CSS as a language). Nevertheless, they have to learn about some rigid syntax and acronyms that are not widely practiced in real life, thus making it harder for the student at the beginning to learn about it.

Using Code versus Software Tools

Initial web developments used to be limited to using simple text editors (like notepad), code the web page and then save the document to display on a browser. It is also called "hand coding" (Kynnin, 2017) in reference entering the HTML lines by typing the code. In cases of hand coding, developers need to remember the syntax (called HTML tags) and the order in which they need to appear for correct display of the content on the web page. If the web page interfaces with other programming languages (or applications) then the developer may need to learn the syntax of the language in order to code it correctly.

Subsequent progress in the software industry led to the development of many tools that do the work for us – the developers. We can drag and drag objects and place it on the screen, type the text similar to what is done in Word Processing software and the software does the job for us. The software tools generate the HTML code for us so we can display the web page on browsers. All this is done without the need to remember syntax, or the hierarchy of commands or even the order of interface with other languages. The software tools that are used in this manner are called "Visual editors", "WYSIWYG editors" or simply "Web Authoring Tools" (Horton, 2006). The WYSIWYG acronym comes from "What You See is What You Get". It references that what see on the screen at development time is the same thing you will see when the page is displayed in the browser.

The question might be asked about this is that if the software can do all this without the need to remember the code, then what will be the reason to stay with coding. The answer to the question depends of the type of the background of the individual who use code versus the background of developers who use visual editors.

Rode (2004) compared the people who created web pages using software tools versus the people who code web pages. Rod used the phrase "Nonprogrammer Web Application Developers" to describe individuals who design web pages without knowledge of HTML or scripting languages. The same study by Rode used the phrase "professional web programmers" to describe people who design web pages through coding. Rode further compared the two groups by saying:

> Professional web programmers are at least able to create interactive applications while non-programmers are so far limited to creating static web sites. Our preliminary studies suggest that these limitations are not due to lack of interest but rather due to the difficulties inherent in interactive web development. Given the right tools and techniques even the nonprogrammers may become web application developers (p. 1055).

The main point derived from the above quotation is that the coder, or people who develop web pages using code, have more in-depth knowledge of the foundation of web design issues as compared to their counter-parts designers who use software tools to develop web pages. It also implies the importance of understanding the foundation upon which web design is typically being built on. This foundation may be more understood when designer understand the "behind the seen code" that display the page. (Kyrnin, 2017) expressed even more support for coding web pages using simple text editors and noted "Web developer that can write HTML using a text editor will be more marketable than one who can use WYSIWYG editor" (p. 1).

In contrast to the view expressed above, proponents of using software tools describe the use of coding to create web pages as a process of "yesteryear" (Bardzell, 2006) or old-fashioned way of doing web pages. Added to that the web authoring tools have increased substantially in number and in ease of use. This made it easier to create more advanced and sophisticated web pages without learning much about coding.

Free Versus Propriety (Paid for) Software

This may sound like a trivial question, after all how many people would consider paying for something (in this case software) if they can get it somewhere else free of charge. Complicating the issue further an increasing number of web development software is being offered free. Furthermore, the functionalities of the free software and their ease-of-use are increasing as well so that in many instances they come at par with licensed (or paid for) software.

A simple search of the web for "Free web authoring tools" yields an enormous number of software available for download without conditions and questions to answer. The downloads are free and can often be installed by simple clicking the download button.

Through our brief search for "free web authoring tools" on the web, we found plenty that supports the contention above. We found one web site lists nine best free HTML editors for web developers. When we examined the list of nine further, we found they are legitimate software that do the job of authoring web pages effectively. Another web site offered a list of ten free cross-platform web authoring tools. Similarly, the list contained legitimate software that do the job without strings attached to them. Some even are supported by major companies. For example, Microsoft offers a free software called "Expression Web" and they also provide free videos, free tutorials and free example files to download for training on using this software.

On the other hand, free is not always free and there some contingents attached to it in cases. Fitzerland and Suzor (2005) Explained about the misunderstanding of "free" software: "Free software is not free because it has no price; it has free because it embodies value that enhance liberty for users and programmers" (p. 3). Fitzerland and Suzor, noted that in order to download the software, users must consent to the licensing agreement in the software. This licensing agreement often stipulates clauses that include the following:

- No use of the software to making profit
- Updates and enhancements to the software need to be shared with the public
- Damages caused by this software (like by accident downloading software) may not hold the software writers responsible for damages
- Credits for additional development must be given to the software

Kennedy (2001) on the other hands expressed a less cautious approach regarding the use of free software. Kennedy noted that free software were made based on the "community sharing" concepts – that is software developed by the community and enhancements are to be return to the community. Nevertheless, Kennedy emphasized the importance of reading the license agreement of the software before beginning for their extensive use.

Teaching Styles

Teaching style can vary from one course to another. But in regards to IT teaching, two styles are common: Explorative and learning by doing. Ye and Kishida (2003) explained about these two styles of learning:

> Explorative learning- This form of learning is similar to the most scientific research (e.g., scientists, practitioners) attempts to find new ways of doing things or of overcoming an existing problem. OSS systems viewed by GNU developers as scientific knowledge to be shared among mankind. Larry Wall started Perl because he ran into a problem he couldn't solve with existing tools, and he wanted to explore a way of doing things better.
> Leaning by doing- In this form of learning, the learners want to deepen their understanding of a certain tasks by actually engaging in practical tasks that allow them to apply their existing knowledge and to perfect their skills (p. 425).

The two styles of learning may apply to any teaching environment in general and may not be limited to one field of study. Despite the generality of both learning styles, a decision can be made regarding their applicability. The preference of using one learning style over the other in web design courses is not a clear cut. If programming (or coding) content is covered extensively in the course, then it may be argued that programming students traditionally learn more by doing hands on programs (Rode, 2004). Explorative learning can be used in cases when the course uses software tool and to show the students few application concepts.

In web design teaching, both approaches can be practiced. The field of web design has a lot of theories, concepts and approaches that can be lectured and explained before the giving of practical examples. Similarly, having the students develop learning by doing has it is benefits to the students as well. In this approach, examples can be given and have the students practice on them. The students can learn from these practices, they can learn from the errors they make and through their repeated exposure to the application.

Teaching Web Design Course Again

The department of Information Systems and Decision Sciences (ISDS) at Eberly College of Business and Information Technology (ECOBIT) - Indiana University of Pennsylvania (IUP) offers a course titled "IFMG201 – Internet and Multimedia". Although the course name does not indicate web design, but the course syllabus was written in broad terms that allows some flexibility in selecting content to teach. The faculty teaching this course (the author of this paper) has received numerous requests to design web sites from non-profit organizations. The faculty decided to cover web design as the main topics for teaching IFMG201. The following factors are important to remember in the consideration for designing our IFMG201 course:

- Most of the students registered in the course come from non-technical majors. Thus, introducing coding or programming concepts in any depth may not seem practical

- The students are required to complete a service learning project. In this, they design web sites for non-profit organizations in the community
- All non-profit organizations that participated this semester in this project, wanted to make their web site mobile ready
- All non-profit organizations want to learn how to maintain the web site on their own once this project is completed and after the commencement of the semester
- The service organizations prefer to have free software installed on their computers to maintain the files on the web sites and make changes when necessary

The remainder of this section explains about the decision of the faculty in regards to the issues that were discussed for teaching this course.

MOBILE FIRST VERSUS MOBILE LATER

The decision by the faculty (author of this paper) is to wait until later to teach about designing the web page for mobile devices. The students enrolled in the course need to gain some knowledge about web design before delving into the coverage of specific devices. Teaching the specifics about making a web page more adaptable to various devices require some initial knowledge about web design. There are various templates that can be downloaded and used to make web pages mobile ready. Nevertheless, in order to use these templates correctly, students need to learn about the essentials of web design. It will help the students more if they learn these essentials of web design at the beginning and then apply work on the issues that focus on designing web sites for mobile devices later in the course.

CSS First versus CSS later

Although it may be a good idea to start teaching CSS from the beginning, but the faculty decided to use an approach different than what is suggested in the literature review. At the beginning, the faculty will be using an already written code of CSS. So it is almost concurrent learning of HTML and CSS at the same time. The main reason that led to the decision of concurrent teaching of CSS and HTML is that HTML5 does not encourage doing any formatting inside HTML document. Every formatting need be included in the CSS document. Even the most widely used web authoring tools emphasize the use of CSS classes in formatting. Any sort of formatting will produce CSS classes by the software instead of the old formatting that was included in earlier versions of HTML. Thus the author decided to teach them concurrently and teach CSS at stages. First, introduce an already coded CSS files and the students can use the pre-written code to work on developing pages. The students at later stages will be required to write their own CSS code after learning the earlier concepts in the course.

Using Code versus using Software Tools

Using software tools is the most appropriate for the group of students that are taking my course. The reason is that most of the students in the course do not have programming background, most do not like to take programming courses and it is prudent to use software tools instead. In some cases, there will be a coverage of explaining about HTML tags especially when applying styles and writing code for divisions in the web pages. Also, it is important to teach some coding when including teaching about responsive web site in order to make the web site more ready for display at various device sizes (like mobile devices).

Using Free Software versus using Commercial software

The decision is to use paid for software for my course. Our university has a campus wide license agreement with Adobe. The agreement allows for installing the software on all campus computes. We have worked on different versions on the same license agreement and the current version of the software suite that we have is Adobe Creative Cloud (Adobe CC). The main software we use for our courses is Adobe Dreamweaver to teach about designing the web pages. We also use Adobe Photoshop for image editing and Adobe Premiere for video editing and Adobe Flash (or Animate) for creating animation for different web pages.

This decision to use this software got a little complicated as the non-profit organizations asked to have free software installed on their computers so they can maintain their web sites. The author decided to continue with teaching Adobe CC for the course. Later on, the faculty can show the students the work of other free web authoring software. The students in turn can show the organizations how they can download, install and work with on the use of the free software.

About the Teaching Style

The author uses the explorative learning style at the beginning of the course and then follows the learning by doing style through the remainder of the course especially at the end when they design web sites for non-profit organizations. Explorative learning style seems to work well with the students at the beginning because it gives them necessary background information so that they can apply the concepts later on. In the time of application, the learning by doing would seem to be more appropriate to use. In the "learn by doing" style, the faculty gives the students work to complete, he gives time to complete the work and his role for this work is to guide the students to find solution and also give help when needed.

SUMMARY

This paper discussed the factors that need to be considered when returning to teaching web design courses. It began by introducing a literature review about the critical factors that need to be addressed for teaching web design courses in light of the most recent advances in the field. It then talked about the decision of the author teaching a course in web design in regards to the factors discussed in the literature review. Table 1 below summarizes the decision that this faculty made in regards to the factors considered for teaching web design courses:

Table 1. Summary of Selection of Paradigms to teach for Web Design Course

Paradigm	**Decision**
Mobiles vs mobile later	Mobile later to give students to learn
CSS first versus CSS later	Concurrent teaching of CSS and HTML
Using code vs software tool Free versus commercial software	Use web authoring software tools
Free versus commercial software	Commercial software through the course and free software for the final and service learning
Explorative vs Learning by doing	Learning by doing throughout most of the course with occasional understanding of code

REFERENCES

Ali, A., & Mensch, S. (2008). Issues and challenges in selecting content for web design courses. *Journal of Issues in Informing Science and Information Technology*, *5*, 209-231.

Bardzell, J. (2006). *Macromedia Dreamweaver 8 with ASP, ColdFusion, and PHP, Training from the source.* Berkley, CA: Macromedia Press.

Ben-Ari, M. (2010). Objects never?: well, hardly ever!. *Communications of the ACM*, *53*(9), 32-35.

Bishop, S., Shuman, J., & Vodnik, S. (2016). *The Web Collection Revealed Creative Cloud: Standard Edition*. Boston: Cengage Learning.

Blansit, B. D. (2008). An Introduction to Cascading Style Sheets (CSS). *Journal of Electronic Resources in Medical Libraries*, *5*(4), 395-409.

Craig, E. M. (2007). Changing paradigms: managed learning environments and Web 2.0. *Campus-Wide Information Systems*, *24*(3), 152-161.

Ehlert, A., & Schulte, C. (2009, August). Empirical comparison of objects-first and objects-later. In *Proceedings of the fifth international workshop on Computing education research workshop* (pp. 15-26). ACM.

Ernst, W. (2002, September 3rd). Cascading Stylesheets. *PC Magazine*, IP1-IP4.

Fitzgerald, B., & Suzor, N. (2005). Legal issues for the use of free and open source software in government. *Melb. UL Rev.*, *29*, 412.

Fowler, S. L., & Stanwick, V. R. (2004). *Web application design handbook: Best practices for web-based software*. Morgan Kaufmann.

Gafni, R., & Barak, O. (2013). Exploring the addition of mobile access to a healthcare services website. *Issues in Informing Science & Information Technology*, *10*, 163-175.

Ganci, A., & McCullough, J. (2015). OCLC on the Responsive Web. *Library Technology Reports*, *51*(7), 44.

Glassman, N. R., & Shen, P. (2014). One site fits all: responsive web design. *Journal of Electronic Resources in Medical Libraries*, *11*(2), 78-90.

Horton, S. (2006). Choosing a web authoring tool. *Web teaching guide.* Retrieved September 13, 2006

Kennedy, D. M. (2001). A primer on open source licensing legal issues: copyright, copyleft and copyfuture. . *Louis U. Pub. L. Rev.*, *20*, 345.

Kim, B. (2013). Responsive web design, discoverability, and mobile challenge. *Library technology reports*, *49*(6), 29-39.

Kyrnin, J. (2017). Pros and Cons of Text Editors. Retrieved February 1st, 2017 from http://webdesign.about.com/cs/htmleditors/i/ai_whytext_2.htm

MacConaonaigh, S. (2007, November). The Best of Both Worlds. *Informatics*, 11-15.

Motes, G. (2012). *HTML5 may provide vital link for friendly future mobile applications*. Army Signal Center Of Excellence Fort Gordon Ga.

Natda, K. V. (2013). Responsive web design. *Eduvantage*, *1*(1) retrieved January 31, 2017 from http://cms.omvvim.ac.in/index.php/eduvantage/article/viewFile/18/pdf

Robertson, A. (2002). Cascading Style Sheets 2.0: Programmer's Reference.(Book Reviews). *Technical Communication*, *49*(3), 363-365.

Rode, J. (2004). Nonprogrammer web application development. CHI 2004 *Doctoral Consortium*. Retrieved October 20, 2006 from ACM Digital Library http://www.acm.org/dl

Smilansky, O. (2016). Retailoring Retail for a Mobile-First World. *CRM Magazine*, *20*(7), 30-33.

Zhu, H., & Zhou, M. (2003, October). Methodology first and language second: A way to teach object-oriented programming. In *Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications* (pp. 140-147). ACM.

Ye, Y., & Kishida, K. (2003). *Toward and understanding of the motivation of open source software developers.* Retrieved March 20, 2007 from ACM Digital Library http://www.acm.org/dl