This study used an online survey of academic library staff in the southeastern United States to explore what web and/or software development skills are used by library staff whose job responsibilities include those areas; where and when such staff members learn these skills; and, for those with Master of Library and Information Science (MLIS) degrees, how useful LIS courses were in acquiring these skills. Survey results reveal discrepancies between the development skills learned through LIS programs and skills used on the job. They also indicate that academic library staff who have an MLIS degree found LIS courses to be the least useful among various instructional methods for acquiring development skills. This research may help guide changes in LIS curricula to better meet the needs of aspiring academic web librarians, help such students discern what course of study will be most useful, and suggest priorities for on-the-job training for web librarians.

Headings:

Academic libraries -- Officials & employees

Library education -- Evaluation

Web development
LIBRARIANS WHO CODE: WEB AND SOFTWARE DEVELOPMENT SKILLS OF ACADEMIC LIBRARY STAFF IN THE SOUTHEASTERN UNITED STATES

by
M. Allen Bell

A Master’s paper submitted to the faculty of the School of Information and Library Science of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in Information Science.

Chapel Hill, North Carolina
April 2014

Approved by

_______________________________________
Ericka Patillo
Introduction

The increasing importance of information technology to libraries in recent decades has brought with it the need for new job skills and job titles. The rapidity of change in technology destabilizes job descriptions as the library profession adapts to the shifting expectations of users in our increasingly technocentric culture. This study investigates academic library staff whose job responsibilities include web and/or software development. These librarians have a great variety of job titles and often work in various departments—perhaps because libraries have recognized a need for librarians with programming skills but not enough time has passed for a standardized role to be established, or perhaps because many librarians in various departments will need to have some degree of programming skills in the future. Library and information science (LIS) programs have recognized the growing demand for IT skills in libraries and have responded in the past decades with an increase in courses offered that focus on developing IT skills such as database management and design, web development, and computer programming.

A great deal has been written about courses offered in LIS programs, changes in skills listed in library job announcements, and the experiences of practicing librarians with the technology in the workplace. The literature concerning the skills and practices of librarians who use IT skills in their work contains many insights, but the studies assessed in the literature review below conflate, in various combinations, the diverse skills of web design; HTML and CSS; content management systems (CMSs); web programming
languages; and compiled programming languages usually associated with desktop or mobile application development. One of the goals of the current study is to disaggregate these related but distinct skill sets in order to achieve a greater understanding of which web and software development skills are currently in demand.

The questions this study seeks to answer are as follows:

RQ1: Where and when do academic library staff whose job responsibilities include web and/or software development learn the skills needed to perform this work?

RQ2: What web and/or software development skills are used by academic library staff whose job responsibilities include web and/or software development?

RQ3: For those with a Master of Library and Information Science (MLIS) degree, how helpful was their LIS education for their web and/or software development work?
Literature Review

Content analysis of curricula
LIS programs have been increasing the number of IT-related courses they offer since at least the 1980s. S. He analyzed the course catalogs of four LIS programs over a fifteen-year period and found a substantial increase in the number of IT-related courses (He, 1999). By spring 2008, 60% of 55 of the 57 ALA-accredited programs (all but one in Puerto Rico and one in Montreal) included an IT course of some sort as part of the core curriculum (Hall, 2009).

Hu performed a content analysis of course offerings of the US News and World Report’s 2014 Best Library and Information Studies Schools for the 2013–2014 academic year (2013). On average, 33% of the courses offered at the 14 top-ranked programs (all ALA-accredited, all but one of which was an iSchool) were IT courses (Hu, 2013). Hu notes that many of the traditional librarianship courses include IT elements, which would give an even higher percentage of courses with some IT component; she did not include this in her data collection process, so she could provide no quantitative data (2013). She further observes that half of the fourteen LIS programs offer IT-related degree programs or have merged with computer science or IT programs (Hu, 2013).

The most common type of IT course offered among the fourteen programs was database design/system management (34% of IT courses across all programs were of this type), followed by web/social networks (28%) (Hu, 2013). This echoes the trend found by Riley-Huff and Rholes in the 2009–2010 academic year in their content analysis of all
57 ALA-accredited programs, with 70% of programs offering at least one course in database design, development, and maintenance, and 68% offering at least one in web design, development, or usability (2011). A total of 47 database courses were offered across all ALA programs, including 7 advanced courses, and 52 web design, development, or usability classes were offered, including 11 advanced courses (Riley-Huff and Rholes, 2011).

Computer programming was ranked at or near the bottom of IT skill priorities across several articles. As of fall 2008, only 2 out of all 57 ALA-accredited programs required incoming students to possess any programming skills, compared to 13 programs requiring database experience and 6 requiring the ability to hand-code a web page in HTML (Kules & McDaniel, 2010); 17% of all 57 ALA-accredited programs offered at least one programming course in the 2009–2010 academic year, with only two advanced courses offered (Riley-Huff & Rholes, 2011); and 12% of IT courses offered in the 2013–2014 academic year at the fourteen top ranking LIS programs in the US focused on computer networks or programming (Hu, 2013).

**Job list analysis**
Analysis of job postings, although a relatively common method of discerning what skills are required in library jobs, has its shortcomings as a methodology. As Cox and Corrall note, “job advertisements do not necessarily give a clear indication of the true requirement or what people employed as a result of the advertisement actually do” (2013). Cox and Corrall further observe that job postings are particularly unreliable indicators for what the position will entail for emerging roles that haven’t yet become well defined in the profession, and that it is difficult to compare results across studies.
because there is no standardized vocabulary in job postings and each researcher usually creates their own coding scheme for the data (2013).

This general difficulty of comparing results certainly applies to the three studies discussed here. It is difficult to compare results on the demand for programming languages and web development skills in job ads because different studies define these terms differently. Mathews and Pardue define web development as including HTML, XML, and writing for the web, while they define programming as including languages such as Java and C++; neither of these definitions mention the most common web programming languages, such as JavaScript and PHP, so it is unclear how frequently these skills occurred in their study (2009). They report a higher percentage of ads demanding web development skills than the other studies reviewed here, but this number (37.75%) was likely inflated by the inclusion in their definition of skills that do not require any specialized IT knowledge, such as writing for the web (Mathews & Pardue, 2009). Tzoc and Millard provide the clearest set of categories, breaking web development and programming skills into five separate categories, though there is some confusing overlap between the category “web application development,” which is defined as including scripting languages, and another called “scripting languages” (2011). Tzoc and Millard’s and Choi and Rasmussen’s studies each show higher percentages of job announcements requiring IT skills than do Mathews and Pardue, because the former two studies focus on “digital” positions, whereas Mathews and Pardue based their results on a sample of all postings on the ALA JobList requiring an MLIS or its equivalent—a broad pool that dilutes the demand for IT skills relative to explicitly “digital” positions (Choi & Rasmussen, 2009; Mathews & Pardue, 2009; Tzoc & Millard, 2011).
The discrepancies between the two studies focused on these “digital” positions merit consideration. Tzoc and Millard found a much higher demand for programming skills than did Choi and Rasmussen, reporting that 24% of job postings in their sample listed programming languages such as Java and C++, and 64% listed scripting languages such as JavaScript and PHP; Choi and Rasmussen found that 10.34% of postings required either of these types of languages (Choi & Rasmussen, 2009; Tzoc & Millard, 2011). The difference in the time span covered by the studies, in the context of an increasingly digital world, could explain some of this apparent increase in demand—the higher values come from a sample of postings from January to December, 2010, while the lower value comes from a sample from 1999 to 2007—but it seems unlikely that such a dramatic leap would occur in the three years separating the two samples (Choi & Rasmussen, 2009; Tzoc & Millard, 2011). The difference between postings for “digital librarian positions” (Choi & Rasmussen, 2009) and “digital library or digital collections” positions (Tzoc & Millard, 2011) also seems unlikely to account for such a significant disparity. The job lists from which each study pulled their job postings do not seem to explain this, either—though one may well question the sampling method of drawing from three job-list sites and three LIS program career pages, as did Tzoc and Millard, compared to the simplicity and validity of drawing from a single prominent job list such as College and Research Libraries News, as did Choi and Rasmussen (Choi & Rasmussen, 2009; Tzoc & Millard, 2011). It is unfortunate that the study with the questionable methodology is also the only study of the group that clearly distinguishes the boundaries of programming languages, web programming languages, and web design (Tzoc & Millard, 2011). Though their numbers are rather high, their rankings seem plausible: web design, including HTML/CSS, ranks highest (72%), followed by scripting languages, which typically
accompany any modern web development project (64%), with other programming languages not commonly used in web projects in a distant last place (24%) (Tzoc & Millard, 2011). From these three studies, it seems sufficient to say that this is the typical ranking of skills in demand for librarians and that these skills are more in demand for librarians who work closely on digital projects than for the broader category of academic library positions in general.

**Web librarians: surveys of current practitioners**

Although job lists offer a convenient source of data to analyze to get a sense of what skills are expected of academic librarians, job announcements do not always accurately predict what work the position will eventually entail. Surveys of current practitioners, in comparison, are “more likely to be accurate about what people actually do,” and thus complement the findings of job-list analyses (Cox & Corrall, 2013).

Two studies have indicated that most web librarians have an MLIS or equivalent degree: 78% of webmasters at 82 ARL libraries reported having the degree in Taylor’s 1998 survey, and 74% out of 61 solo webmasters in medium-sized academic libraries had it in Kneip’s study (Kneip, 2007; Taylor, 2000). Not all web librarians with an MLIS felt that their library program had prepared them well for working with the web. In Taylor’s 1998 survey of library webmasters at 82 ARL libraries, only 28% of respondents felt that their LIS education had prepared them adequately for their work (Taylor, 2000). Seventy-eight percent of web librarians who responded had an MLIS degree; 83% of this group had graduated in or earlier than 1993, the year the Mosaic web browser popularized the web (Taylor, 2000). Of those who felt that their LIS education had inadequately prepared them to be library webmasters, 78.38% attributed this to the fact that the web either did not exist yet or was too new; 21.62% said that the web had already
existed but their program had not offered any IT or web courses (Taylor, 2000). Several years later, 45% of respondents to Kneip’s survey said they were “satisfied” or “very satisfied” with LIS courses; however, it is not clear whether the 55% who reported being unsatisfied had taken any courses in web development, or whether their programs had offered any such courses (2007). The increased reports of satisfaction seem to indicate that the inclusion of more IT-related courses (discussed above in the section on curriculum analysis) has been helpful for those who would become web librarians.

The addition of IT-related courses to the LIS curriculum can serve as a valuable supplement for the many aspiring librarians with little or no technical background. Though Connell found that over one-third of web librarians surveyed had some prehire training in web development through one or more undergraduate or graduate courses or professional workshops, both Taylor and Kneip reported that only one-tenth of web librarians who responded to their surveys had a degree in computer science or a related IT field (Connell, 2008; Kneip, 2007; Taylor, 2000).

Self-instruction is widespread among web developers both in and out of the library profession. In a survey of web developers outside of libraries, nearly all (95.2%, N=270) respondents said that some of their web skills were self-taught (Rosson, Ballin, & Rode, 2005). Surveys of web librarians have found similar numbers: over 80% of Taylor’s respondents taught themselves computer science and/or HTML, and about 90% of Kneip’s respondents included self-instruction as part of their web development education (Taylor, 2000; Kneip, 2007).

Whether one is self-taught, has taken only a course or two in web development, or has a degree in computer science, any foundation of web and programming knowledge needs to be updated constantly. This explains in part why the predominant interest in
continuing education across all types of librarians is in technology (91%, N=2194), as demonstrated in Marshall et al.’s ambitious career study of all graduates of LIS programs in North Carolina from 1964–2007 (2009). Most academic libraries appear to offer satisfactory opportunities for continuing education in IT skills, with over two thirds in Bosque and Lampert’s survey expressing satisfaction with their employers in this area and reporting that they take classes and workshops (2009). It seems that courses and workshops do not suffice to keep up with technology, however, as almost all (92.4%) respondents reported pursuing self-instruction outside of work hours (Bosque & Lampert, 2009). Connell corroborates this theme of self-motivated learning, citing the most common type of continuing education mentioned by respondents to her survey as self-instruction through books, articles, and websites (2008).

It is common for a library to have only one or two librarians with responsibility over the library website; among the respondents to Taylor’s survey, about half of library webmasters worked alone and about one-quarter shared responsibilities with one other librarian (2000). In most reported cases, the primary web librarian spends fewer than 20 hours per week on web-related work (Kneip, 2007; Taylor, 2000). Even though there is often only one librarian working part-time on web projects, web work is usually conducted in collaboration with a web committee (Bundza, Meer, & Perez-Stable, 2009; Hendricks, 2007; Taylor, 2000). The most common size for web committees is around eight members (Bundza et al., 2009; Taylor, 2000). It is usually such committees rather than the library webmaster who set the library’s web policies (Hendricks, 2007). Taylor writes that nearly 90% of respondents to her survey “preferred to work with a committee,” but notes that the benefits of stakeholder feedback and fresh ideas from
different perspectives are counterbalanced by the extra time required to reach consensus (2000).

Mitchell argues that many libraries conflate the roles of web editor and web developer, and that web developers whom employers require to maintain the content of library sites do not have enough time remaining to focus on web development (2011). Mitchell suggests that reference or outreach librarians could take charge of web publishing, and let the developer focus on the technical aspects of the sites (2011). CMSs enable nonspecialists to author and edit web content with a minimal learning curve, and have been found to improve accessibility on library websites (Comeaux & Schmetzke, 2013; Cox & Corrall, 2013). Though web librarians often continue to write for their library’s website, Bundza et al. stated that almost all (95%) respondents to their survey said that subject librarians contributed content as well (2009). Responsibility for content is often dispersed among several people in the library; Connell’s 2012 survey of academic-library web managers showed that the most common number of authors of library web content was between two and five (2013).

It is important to select a CMS that fits the needs of the library. Library IT staff can perform a key role in procuring a library-appropriate CMS (Connell, 2013). Connell found that libraries at the majority of large institutions (66%) and research institutions (58%) do not use the same CMS as their parent institution, whereas smaller institutions and those offering only master’s or baccalaureate degrees usually do (2013). Unsurprisingly, librarians at libraries that have representation on committees for selecting campus-wide CMSs are more satisfied than those at libraries without such representation (Connell, 2013). It is probably true that libraries at larger institutions are more likely to have internal IT personnel and/or web or systems librarians than libraries at smaller
institutions. Having such staff may help libraries get a seat at the table when their parent institutions select a CMS; one of the respondents to Connell’s survey commented, “Now that we have a dedicated web services position, the library is considered a ‘power user’ in the CMS and we are often part of the campus wide discussions about the new CMS and strategic planning involving the campus website” (2013). In fact, Connell’s unpublished data show that libraries with their own IT personnel are slightly more likely to have a seat at the table for campus wide CMS committees (R. S. Connell, personal communication, January 17, 2014).

Multiple studies over the past several years have demonstrated an increase in popularity of CMSs for academic library websites. Connell’s September 2006 survey, which included a broad swath of 110 academic libraries, from colleges offering associate’s degrees to institutions with doctorate programs, showed about a quarter of institutions represented in the study using a CMS (2008). Bundza et al.’s survey of web librarians in spring of 2008 found that only slightly more than half of libraries surveyed used a CMS (2009). Connell’s 2012 survey, conducted four years after that of Bundza et al., indicated an increase to 63% of libraries using a CMS (2013).

A range of results were found for the percentage of respondents who have experience with programming languages. The variation is perhaps attributable in part to differences in the populations being studied, or to the framing of the question. Bundza et al. reported that 18% (N=116) respondents mentioned programming in their job, but this number is likely impacted by the mixture in their sample of web librarians and reference librarians; they mention that more web librarians than reference librarians responded to the survey and that some respondents filled both of those roles, but they do not provide the statistics for this (2009). Hendricks reports that 13.3% of respondents reported
“coding” as part of their web duties, but 68.3% listed “maintain the web site,” an ambiguous framing that could include web programming languages such as JavaScript and PHP (2007). Bosque and Lampert provide a low figure for use of programming languages on the job (10.5%), but their population was defined as librarians with less than nine years of experience, with no focus on web librarians (2009). Kneip’s study reports a larger number, citing about a third of respondents who described themselves as having “some experience” or being “very experienced” with web programming languages; Kneip further asked about experience levels with specific languages, including PHP, Perl, ASP, and ColdFusion, with JavaScript conspicuously absent from the list (2007). Connell’s survey is the only study reviewed here that asked specifically about JavaScript; she found that 18.2% (N=110) of respondents said that all members of their web team needed to have JavaScript skills (2008). As these studies collectively demonstrate, the shifting of terms, technologies, and priorities in the field poses a challenge for continuing research into these job skills. The state of things will likely improve as the role and relevance of connective technology to library positions becomes more settled.
Method

Participants
The purpose of this study is to survey academic library staff in the southeastern United States who use programming languages to develop websites, web applications, or desktop applications as part of their job responsibilities.

Carnegie Classifications
Institutions were selected based on the 2010 Carnegie Classification of Institutions of Higher Education basic classification lists; several Carnegie basic classifications were placed into two broader categories:

1. Research institutions, defined for this study to include the Carnegie basic classifications: Research Universities (very high research activity), Research Universities (high research activity), and Doctoral/Research Universities.

2. Master’s institutions, defined for this study to include the Carnegie basic classifications: Master’s Colleges and Universities (larger programs), Master’s Colleges and Universities (medium programs), and Master’s Colleges and Universities (smaller programs).

The current study excludes institutions that do not offer advanced degrees. Institutions that offer advanced degrees and conduct research tend to be larger and better-funded and to have libraries with larger staff, with the corollary of an increased likelihood of libraries with internal web or software-development personnel—the target population for this
study. Private institutions were excluded because “funding and staffing at private institutions vary widely, making them less standardized than public institutions, which could potentially skew survey results” (Kneip, 2007).

**Defining the southeastern United States**

To limit the size of the respondent pool, the study focuses on librarians working in institutions in the southeastern United States. There is no single official listing of states to be included in this region; this study uses the definition offered by the Association of American Geographers. They define the southeastern United States as including the following: Alabama, Florida, Georgia, Kentucky, Mississippi, North Carolina, South Carolina, Tennessee, Virginia, and West Virginia (SouthEastern Division of the Association of American Geographers, n.d.).

**Selection of participants**

Upon compiling a list of research and master’s institutions in the southeastern United States, the researcher examined the staff directory of the library of each institution for any staff with job titles that appeared to possibly involve web or software development. The job titles for such positions vary greatly; examples include “web services librarian,” “applications analyst,” “head of user experience,” and “emerging technologies librarian.” Because it is not always evident from the job title whether or not a position includes web or software development, the researcher chose to err on the side of inclusion when selecting potential respondents, and to filter out the false positives with the first question in the survey. The survey begins with the question, “Do you use programming languages to develop websites, web applications, or desktop applications in your library job?” (See Appendix B for a copy of the survey). The Qualtrics survey software (discussed below) sent any respondents who answered “no” to this question directly to a brief thank-you
message ending the survey. This wording is intended to exclude library staff who use programming languages but only for systems administration rather than web or software development, and library staff who use HTML/CSS or a CMS to create or modify web pages but who do not use web programming languages such as JavaScript, PHP, or Ruby.

In cases where no job titles suggest web or software development, a library administrator—or whoever seems to be the person most likely to know who, if anyone, in the library does web or software development for the library—was contacted.

**Survey instrument**
The survey was conducted using Qualtrics, a web-based survey tool. UNC–Chapel Hill provides free access to this software to students and faculty. Among other features, the software allows the survey designer to use conditional logic to determine which questions should be presented to each participant based upon their answers to previous questions.

**Analysis**
Descriptive statistics were employed to present the data harvested from the survey results.
Results

Recruitment
The survey was sent out in March of 2014 to 241 academic library staff members with titles that suggested to the researcher that their positions entailed some amount of web or software development. Out of this group, 95 responded. Of the 95 who took the survey, nine answered “no” to an introductory question designed to filter out anyone who did not meet the criteria for inclusion in the study; these subjects were routed directly to the end of the survey, and no further questions were presented to them. Of the 84 respondents who answered “yes” to this question, one did not answer any questions after this initial filtering question, leaving 83 who completed the survey. The resulting response rate was 34 percent.

The current study had a smaller percentage of respondents with an MLIS than did the studies discussed in the literature review above. Taylor found that 78% of webmasters participating in her study had an MLIS; Kneip reported a similar rate (74%) in his study of solo webmasters at midsized universities (Kneip, 2007; Taylor, 2000). The current study, in contrast, shows 59% of respondents with an MLIS (Table 1). This may relate to differences in study population: the two earlier studies focused specifically on webmasters, defined by Taylor as “someone whose responsibilities may have included,

<table>
<thead>
<tr>
<th>MLIS</th>
<th>MLIS (%)</th>
<th>No MLIS</th>
<th>No MLIS (%)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>49</td>
<td>59%</td>
<td>34</td>
<td>41%</td>
<td>83</td>
</tr>
</tbody>
</table>

Table 1. Respondents with or without an MLIS degree
but were not limited to, Web site policy development, editorial oversight of content
and graphics, organization of files and directories, page maintenance, and user support”
and by Kneip as “an individual who has a significant level of oversight for a library’s
Web site and is actively involved in consistent updates and maintenance for the library’s
Web site” (Kneip, 2007; Taylor, 2000). The current study has aimed to include any
library staff that use web or software development skills in their work.

**RQ1: Where and when do academic library staff whose job responsibilities
include web and/or software development learn the skills needed to perform this work?**

Figure 1 shows that most respondents with an MLIS reported having a bachelor’s in the
humanities and social sciences (69%), whereas most without an MLIS have a bachelors
in computer science or another STEM field (57%). Among those without an MLIS, four
said they did not have a bachelor’s degree; these four were excluded from the calculation
of percentages in Figure 1. The four respondents without a bachelor’s reported gaining

![Figure 1. Respondents’ bachelor’s degrees, listed by discipline](image_url)
their web and/or software development skills through computer-science courses as well as self-instruction through books and websites.

Of the 17 respondents with a bachelor’s degree in computer science, 15 (88%) first started learning computer programming when they were 18 or younger. Among the 63 respondents who do not have a bachelor’s in computer science, 28 (44%) first learned programming when they were 18 or younger, a much lower percentage than is found among those with a bachelor’s in computer science.

For both those with and those without an MLIS, the most common age range to start learning computer programming was from 12 to 18 years (Figure 2), though a slightly higher percentage of non-MLIS respondents reported learning to code in this age bracket than that for MLIS respondents. The largest differences between those with and those without an MLIS were found at the extremes: non-MLIS respondents were 14 percentage

![Figure 2. Age at which respondents first learned computer programming, excluding HTML/CSS](image-url)
points more likely to have learned programming when they were under 12 than respondents with an MLIS, while MLIS respondents were 15 percentage points more likely than non-MLIS respondents to have first learned to code after the age of 30.

Figure 3. Years of web development or programming experience in a non-library job

Those without an MLIS often have experience with programming and/or web development outside of libraries (Figure 3), with most (70%) having four or more years of non-library work experience. Most (57%) of those who do have an MLIS have never held a non-library position involving programming or web-development skills. Out of these 28 with an MLIS who have never held such a non-library position, 20 (71%) have pursued some form of continuing education through professional workshops or courses on web development, a slightly higher rate than the total percentage across all respondents who indicated that they had attended workshops or courses in web development (63%, N=83), and 18 percentage points higher than non-MLIS respondents (Figure 4).
RQ2: What web and/or software development skills are used by academic library staff whose job responsibilities include web and/or software development?

Across both MLIS and non-MLIS respondents, the most frequently cited skill was HTML/CSS. A comparison of the usage rates of various development skills among MLIS and non-MLIS respondents shows a general trend for those without an MLIS to use more such skills than those with an MLIS: with the exception of HTML/CSS and CMSs, higher percentages of non-MLIS respondents reported having used all of the skills, compared to MLIS respondents (Figure 5). Setting aside the Other category, which elicited a miscellany of responses that cannot be counted as one type of skill, the skill with the lowest percentage of non-MLIS respondents still had more than a quarter who reported using that skill. By contrast, fewer than a quarter of MLIS respondents said they had used frameworks, scripting languages such as Perl and Python, and compiled
Figure 5. Skills used in past four weeks, MLIS vs. no MLIS

Table 2. Comparison of ranking of skills listed in Figure 5., MLIS vs. no MLIS

<table>
<thead>
<tr>
<th>Skills used in last four weeks</th>
<th>MLIS ranking</th>
<th>No MLIS ranking</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML/CSS</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>CMSs (e.g. WordPress, Drupal, Joomla)</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Database management (e.g. MySQL)</td>
<td>3</td>
<td>2</td>
<td>-1</td>
</tr>
<tr>
<td>Server-side scripting (e.g. PHP, Ruby)</td>
<td>4</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>Client-side scripting (e.g. JavaScript)</td>
<td>5</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>Unix command line</td>
<td>6</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>XML</td>
<td>7</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>JSON</td>
<td>8</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>Frameworks (e.g. Django, Rails, Symfony)</td>
<td>9</td>
<td>11</td>
<td>2</td>
</tr>
<tr>
<td>Other scripting languages (e.g. Perl, Python)</td>
<td>10</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Compiled languages (e.g. Java, C++)</td>
<td>11</td>
<td>9</td>
<td>-2</td>
</tr>
<tr>
<td>Other</td>
<td>12</td>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>

programming languages. All of the core web-development skills (client- and server-side scripting, database management, CMSs, and HTML/CSS) found strong representation across both groups of respondents.
Ranking the percentage of reported use of each skill in the past four weeks reveals that the relative importance of skills is very similar for those with and without an MLIS. The only two differences are that CMSs were used by more respondents with an MLIS than was database management, while the reverse is true for those without an MLIS, and similarly, the relative rankings for frameworks and compiled languages are flipped for those with an MLIS and those without (Table 2).

Respondents without an MLIS were more likely to have worked on native mobile apps or desktop apps than those with an MLIS, as can be seen in Figure 6. This goes logically with the higher percentage of non-MLIS respondents who use compiled programming languages, which are generally required for building non-web applications (see Figure 5).

Ten respondents reported working on other web projects in the past 12 months (Figure 6). Of the eight respondents who explained what projects they had worked on, the most common type of response was some variety of nonpublic web work (“less public
admin-type sites,” “Digitization studio workflow tracking system,” “ILS ‘hacking,’ white hat”

As Figure 7 shows, a majority of those with an MLIS reported spending fewer than 20 hours in the previous week on web and/or software development (65%), whereas a majority of those without an MLIS reported spending more than 30 hours on development work the previous week (56%), and over three quarters spent 20 or more hours (76%). This agrees with Kneip’s study that indicated that web librarians usually have web-related work as just one among many other responsibilities and that most report spending 20 hours or less per week on web-related work (2007). It’s worth noting that 17 (35%) respondents with an MLIS spent more than half their time the previous week on coding, which is a significant minority.

Figure 7. Hours spent on web development or programming the previous week
RQ3: For those with an MLIS degree, how helpful was their LIS education for their web and/or software development work?

Figure 8. Number of LIS courses taken that were primarily focused on web development, programming, or database management

Figure 8 shows that the majority of respondents with an MLIS degree (61%) took between one and three courses through their LIS program that were focused on web development, programming, or database management.
The skills with the greatest difference between what respondents use and what they learned through their LIS programs are CMSs, JavaScript, and the Unix command line (Figure 9). Also notably underrepresented as a skill learned through LIS programs was JSON, cited by 17 respondents with MLIS degrees as a skill used in the past four weeks but learned by none of the respondents through their LIS programs. This is probably due to the relative newness of the JSON standard, first officially specified as IETF RFC 4627 in 2006, though it had been in use informally since 2001 (Zakas, 2012).

The skill ranked fourth by difference, HTML/CSS, stands out as the only one that a majority of respondents with MLIS degrees reported having learned through their LIS programs, but because nearly all such respondents use HTML/CSS in their work, it nonetheless ranks near the top of skills apparently under-taught in LIS programs. Of the 28 who did not learn HTML/CSS through their LIS program, only three received their MLIS degree prior to the release of the Mosaic web browser in 1993. Unlike those
respondents to a 1998 survey who received their LIS prior to the advent of the web, the
25 other respondents in the current study who did not learn HTML/CSS through their LIS
programs cannot say that the reason was because the web did not exist yet (Taylor, 2000).
However, 11 (44%) of these 25 first learned HTML/CSS when they were 18 or younger,
so they may not have learned HTML/CSS through their LIS programs simply because
they already possessed that skill.

The least-used skill among respondents with an MLIS was compiled programming
languages such as C++ and Java. This finding suggests that the low priority that LIS
programs have given computer programming (as found by several studies discussed in
the literature review above) may be appropriate (Hu, 2013; Kules & McDaniel, 2010;
Riley-Huff & Rholes, 2011). However, due to the varying definitions of computer
programming employed across studies, this point is not entirely clear.

LIS courses did not fare very well in a question that asked respondents to rate the
usefulness of various instructional methods in acquiring their web and/or software
development skills (Appendix B). Among survey respondents with an MLIS, LIS courses
were rated the least useful, with a mean score of 2.63 on a scale of 1–4, where 1 = “not
useful,” 2 = “somewhat useful,” 3 = “useful,” and 4 = “very useful” (Table 3). The
methods with the highest mean usefulness scores were self-teaching with websites (3.71)
and with books (3.56).

<table>
<thead>
<tr>
<th>LIS courses</th>
<th>Workshops</th>
<th>Books</th>
<th>Computer science courses</th>
<th>Websites</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.57</td>
<td>3.03</td>
<td>3.18</td>
<td>3.27</td>
<td>3.52</td>
<td>3.56</td>
</tr>
</tbody>
</table>

Table 3. Mean usefulness ratings given by respondents with MLIS degrees
Figure 10 shows the number of respondents with MLIS degrees who rated the usefulness of various instructional methods, and how many respondents assigned each usefulness rating to each method. LIS courses were deemed “not useful” by three respondents, and were the only instructional method that received this ranking. The three who rated LIS courses as “not useful” all received their MLIS degrees in the past decade (2004, 2009, and 2012), so the negative rating cannot be attributed to older LIS programs that had not yet increased their IT-skill course offerings.

![Figure 10. Usefulness of instructional methods in acquiring web and/or software development skills (Respondents with MLIS)](image_url)
There was not a pronounced difference in the usefulness rating of LIS courses for acquiring IT skills across groups who had taken different numbers of courses focused primarily on web development, programming, or database management (Figure 11). Due to the small number of responses to this question ($N=35$), it is not clear whether a higher response rate or a larger sample would have revealed more substantial differences based on the number of IT courses taken. Further research is needed to discover whether MLIS graduates who take four or more IT courses through their LIS programs express markedly more positive perceptions of the usefulness of those LIS courses.

![Figure 11. Usefulness of LIS courses, grouped by number of IT courses taken through LIS program (MLIS respondents only)](image-url)
Discussion

Tepid ratings for LIS courses
LIS courses were rated by respondents with MLIS degrees as only “somewhat useful” on average. This result aligns with previous research: 28% of respondents in a 1998 survey said that their LIS education prepared them adequately for their work as library webmasters, though 54% (N=54) of respondents to the survey attributed this to the fact that the web did not exist yet when they were in their LIS program (Taylor, 2000). Kneip portrayed a somewhat lower rate of dissatisfaction among academic library webmasters in his 2007 study, citing 55% (N=24) of respondents who rated LIS courses as of little or no use in acquiring web development skills. The language rating scale used in the current study differs from that used by Kneip, but they are both four-point scales of usefulness, and if one compares the percentage of respondents who marked the ratings on the bottom half out of the four-point scale from each study, the current study found a dissatisfaction rate (54%, N=35) similar to the 55% reported by Kneip (2007). Though three studies over the course of 16 years are not sufficient to reveal a clear trend, the slight majority of respondents who held lackluster opinions of the usefulness of their LIS education for acquiring web development skills may indicate a problem that should be considered in planning changes to LIS curricula.

The differences in language used for the four-point rating scales in Kneip’s study and in the current one may provide some support for those who feel more optimistic about LIS programs. Kneip’s scale reads “1 = No use, 2 = Of little use, 3 = Somewhat useful,
4 = Very useful,” whereas the scale used by the current study lists “1 = Not useful, 2 = Somewhat useful, 3 = Useful, 4 = Very useful;” the current study assigns “somewhat useful” to the second-lowest rating, whereas Kneip’s study assigns the same phrase to the second-highest rating. The analysis in the paragraph above compares the results on the numerical scale, looking at the responses on the bottom half of each scale. But if one compared the responses that read “somewhat useful” or better in the current study (i.e., the three most positive ratings out of the four-point scale), one would find an increase from 45% in 2007 to 91% in the current study. This more-optimistic interpretation seems unlikely, however, when one considers that the usefulness rating for LIS courses in the current study was the only instructional method to receive any ratings of “not useful,” and received a larger percentage of the lukewarm “somewhat useful” ratings than any of the other learning methods (46% for LIS courses, followed by computer science courses at 27%).

This data contributes a question to the debate about the role of IT education in the future of LIS programs. Should LIS programs aspire to offer courses that are as useful for acquiring web and/or software development skills as are courses offered through computer science programs? Or should LIS programs instead form partnerships with computer science programs to allow LIS students to take courses through campus computer-science departments? Many iSchools do just that; 7 out of the 14 highest-ranking LIS programs in the 2013 *US News and World Report* offered “IT related degrees” or had merged with computer science or IT programs (Hu, 2013). A future study could survey graduates of these 14 programs to see whether programs that have more strongly identified themselves with computer science and IT are rated as more useful for acquiring web development skills than are less IT-oriented LIS programs.
The instructional methods that received the highest usefulness ratings from graduates of LIS programs centered around self-education through websites and books. Free text responses elucidate some of the specific types of online resources that respondents have used. Of the 17 respondents who cited “other” instructional methods, six wrote about various types of self-motivated learning through materials available online: two respondents noted that Lynda.com, a subscription-based library of online training videos, was very useful; two mentioned reading source code and rated this as useful; and two cited online documentation and reference websites, though they did not assign usefulness ratings to these resources. These three pairs of respondents do not provide statistically significant data, but they do give a qualitative sense of the types of web-based learning materials that respondents use.

Seven out of 17 (41%) of respondents who cited “other” instructional methods mentioned learning from other people—colleagues, friends, or other professionals met online or at conferences. Five of these seven rated other people as very useful; two did not offer a usefulness rating. Two others mentioned hands-on learning, one rating it as useful and the other as very useful. LIS programs could, and many probably do, offer this type of learning experience through independent studies and internships for class credit. It is possible that some respondents had such learning experiences through their LIS program but did not include them in their evaluation of the usefulness of LIS courses; one respondent commented that he had learned IT skills through a research assistantship in his LIS program, but he did not provide a usefulness rating for this experience. These comments reaffirm the importance for students to gain work experience as part of their LIS studies.
Teaching skills that practitioners use

Despite the differences in how IT skills are categorized across studies, certain basic patterns emerge. The rankings of demand for skills in three of the articles discussed in the literature review corroborate the findings of the current study: these studies found HTML/CSS to be in the greatest demand, followed by web programming languages such as JavaScript and PHP, and distantly trailed by programming languages not typically associated with web development (Hu, 2013; Kules & McDaniel, 2010; Riley-Huff & Rholes, 2011). The current study also found that demand for HTML/CSS skills is nearly universal for librarians involved in web development and programming, with nearly two-thirds of librarians using web-scripting languages such as PHP and JavaScript; by contrast, library professionals with an MLIS rarely use compiled programming languages, which are typically associated more with native mobile and desktop applications than with web development.

The greatest discrepancy between skills used on the job and skills learned through an LIS program is found in CMSs (Figure 9). However, it is unclear what LIS programs should do with this information. Knowledge of one CMS is not necessarily generalizable to other CMSs, and such systems undergo constant changes, which makes it impractical to develop a course around them.

CMSs such as WordPress have at least two layers. There is the end-user interface for authoring, editing, and managing content, which is not very difficult to learn for anyone familiar with word processors. The user can also choose to edit the content and presentation by using an HTML editor or changing the CSS file, but these skills are not distinct from general HTML/CSS skills and would not merit special instruction either. The complexity for WordPress arises in theme and plugin development, which requires
knowledge of PHP as well as the WordPress framework (Williams et al., 2013). If an LIS program were to offer a course on CMSs, this is the area that would be most logical to focus on, perhaps as an advanced course after a student has already learned the relevant web technologies (HTML/CSS, JavaScript, PHP). But because it is not clear which aspect of using CMSs respondents to this survey had in mind, it is not clear whether learning to program aspects of CMSs such as WordPress plugins or Drupal modules is in demand, or whether most respondents merely use a CMS to author and manage content.

The second and third largest gaps between skills used on the job and skills learned through an LIS program are found in client-side scripting languages such as JavaScript and the Unix command line (Figure 9). Sixty-one percent reported using client-side scripting languages in the past four weeks, compared to 8% who reported having learned this skill through an LIS program; for Unix, the ratio was 57% used versus 14% learned (N=49). In contrast to CMSs, which vary across platforms and change rapidly as noted above, JavaScript and Unix are stable technologies that have been important skills for web developers and programmers for many years. JavaScript has a long history of web browser support, beginning with support for JavaScript 1.0 with the release of Netscape Navigator 2 in 1995; Unix was developed in Bell Labs from 1969 to 1974 (Campbell-Kelly & Aspray, 2004; Zakas, 2012). The results of this study suggest that JavaScript and Unix should receive greater emphasis, replacing instruction in compiled programming languages. Nineteen percent of respondents reported learning such languages through their LIS program, while only 4% with MLIS degrees said they had used them in the past four weeks (Figure 9). The problem is not that nobody in working in libraries actually uses compiled programming languages like Java and C++; rather, it is predominantly
library staff who do not have MLIS degrees who use those languages: 35% of those who don’t have an MLIS compared to 4% of those who do. Because those who don’t have an MLIS appear to be the ones using compiled programming languages in the library, it would seem to be more beneficial for MLIS students to learn to program using a language they are likely to use in their career, such as JavaScript.

**When librarians learn programming**
The results in Figure 2 show that respondents without an MLIS degree were more likely than those with a library degree to have learned programming when they were young, while those with an MLIS were more likely to have first learned to program as adults. One explanation for this could be that those without an MLIS had an interest in programming prior to working in a library, and that such IT skills may have been why the library hired them, whereas many respondents with an MLIS may have been initially interested in librarianship and only began learning programming through their LIS program or library job. However, nearly half (45%) of respondents with an MLIS reported learning to program when they were 18 or younger; the primary finding displayed in Figure 2 is not that librarians don’t learn to code until they are adults, but that those without an MLIS usually already had started learning to program by their mid-twenties. To further emphasize that many MLIS respondents also learned programming when they were young, five of the 17 respondents with a bachelor’s in computer science also hold an MLIS degree, four of whom learned to program when they were 18 or younger. This follows the larger trend: nearly all respondents with a bachelor’s degree in computer science (88%, N=17) started to program when they were 18 or younger. This seems unsurprising; it makes sense that students would be more likely to choose to study computer science if they already had experience with it and knew that it interested them.
But success in informal self-instruction may sometimes also influence a person to not pursue that skill through formal education: one respondent commented, “My LIS program taught basic HTML but I had learned it before then.” If LIS programs want to encourage the development of these skills, they might consider promoting courses that teach them to students who did not enter the program with prior experience in them. Surveying students entering programs about their experience in this realm would provide useful data for advising and program planning.

**Limitations**
Due to the lack of consistent job titles for web librarians, the selection of subjects was based on the researcher’s judgment as to whether each job title suggested that the position might entail web and/or software development. The first question in the survey served to filter out anyone who had been invited to participate but who did not “use programming languages to develop websites, web applications, or desktop applications in your library job”; trusting this question to filter out false positives, the researcher chose to err on the side of inclusion rather than risk excluding any potentially viable participants. Nonetheless, there were probably some librarians who did fit the study’s criteria but whose job title did not indicate this to the researcher. Furthermore, erring on the side of inclusion probably had a negative impact on the response rate.

It is possible that a greater number of respondents without an MLIS degree would have participated in the study had the invitation to take the survey been phrased differently. The email invitations to participate in the study described the target survey population as “academic librarians.” Upon receiving several emails from willing participants who did not believe they met the criteria for the study because they were not technically librarians, or because they did not have an MLIS degree, the researcher
realized that a more accurate phrase for the target population would be “academic library staff.” Unfortunately, there was not enough time to get permission from the Institutional Review Board to make the needed changes to the reminder email; some potential participants may have chosen not to participate because they did not believe they were met the criteria for inclusion in the study.
Conclusion

This research has endeavored to discover what skills are used by academic library staff who have web and/or software development as part of their job responsibilities; where and when they learned those skills; and, for those who have an MLIS degree, how useful LIS courses were for acquiring the development skills they use in their work. Survey results revealed that respondents without an MLIS are more likely to use compiled programming languages and work on native mobile and desktop applications than are those with an MLIS. Across all respondents, regardless of education background, the most common age range to have first learned computer programming was from ages 12 to 18, though it was more common for respondents with an MLIS to have learned programming as adults than those without MLIS degrees. This may be because those without an MLIS degree chose to pursue a career in web or software development and happened to find a position in a library, whereas those with an MLIS chose to become librarians found the increasing demand for IT skills in libraries created an incentive to acquire web or software development skills, either through LIS courses or through learning such skills on the job. Discrepancies between the skills that respondents with MLIS degrees reported using and those they reported having learned through LIS programs indicates that LIS programs should shift away from teaching compiled programming languages such as Java and C++, and place more emphasis on skills such as JavaScript and the Unix command line.
Future research could explore the reasons behind the lukewarm usefulness ratings that respondents of this survey gave to LIS courses for acquiring web and/or software development skills. Such work could build on research into the informal learning strategies of web designers and other professionals who acquire programming skills on the job, and apply this research to the academic library workplace (Dorn & Guzdial, 2006, 2010; McCartney et al., 2010; Rosson et al., 2005; Zander et al., 2012). Future surveys could probe the reasons underlying the dissatisfaction with LIS courses that respondents of this survey expressed: were the LIS courses not useful because the skills taught were not the same skills required in the workplace? Were the instructional methods unsatisfactory? Or do web librarians simply prefer informal education through web resources and instructional books over formal courses offered through LIS programs?

The results of the current study may prove helpful to future LIS students who aspire to careers in web librarianship as they try to ascertain what skills they should cultivate to succeed in this emerging field. Given the increasing relevance of connective technologies to librarianship, it may offer insights for LIS students without special interest in web librarianship as well. The study may also help inform changes in curricula for LIS programs seeking to better meet the needs of students who are interested in pursuing a career that combines librarianship and an interest in computer programming or web development. As web librarians’ roles become more clearly defined, we can hope that an attendant clarity in curriculum planning will result, and that future survey respondents will express more satisfaction with their LIS courses’ relevance for web development.
References


SouthEastern Division of the Association of American Geographers. (n.d.). SouthEastern Division of the Association of American Geographers. *SouthEastern Division of


Appendix A: Emails Sent to Participants

Subject: Invitation to participate in a study of librarians who code

Dear [Firstname],

I hope you’ll consider participating in a survey of academic librarians whose job responsibilities include web and/or software development. Based on your job title, I believe you are eligible to take part.

This study aims to help LIS programs and their students understand what skills are needed to succeed in jobs similar to yours. LIS programs have begun to offer more courses focused on IT skills such as web development and programming, but it is not clear how helpful those courses have been for practitioners like yourself. Your responses will help improve understanding of these issues.

The survey consists of 20 questions and should take 10 to 15 minutes to complete. It can be accessed with the following link, which is unique to your email address:

[link to survey]

If you choose to participate and would like to see the results of my research, it will be made available through the Carolina Digital Repository later this year.

Thank you for your time!

Allen Bell
2014 Candidate for Masters of Science in Information Science, UNC Chapel Hill

------------------------------------------------------------------------------------------

Subject: Reminder: Please participate in a study of librarians who code

Dear [Firstname],

You recently received an invitation to participate in a study of academic librarians whose job responsibilities include web and/or software development. It looks like you haven’t submitted a survey yet. If you would like to participate, the survey will only remain open for three more days.

[link to survey]

A note on privacy: your responses will be reported in anonymous aggregate statistics, and the survey software will report only your IP address to me, so I will not be able to pair your responses with your identity. On that note, it is only the Qualtrics survey software
that knows that you haven’t submitted a survey yet; I do not personally know that you haven’t taken the survey.

The survey consists of 20 questions and should take 10 to 15 minutes to complete. By participating, you will contribute to the future of your profession by shedding light on the skills required in the ever-evolving work environment of the web/software developer librarian.

Please contact me with any questions you have about the survey at bellma@live.unc.edu.

Best,

Allen Bell
2014 Candidate for Masters of Science in Information Science, UNC Chapel Hill
Appendix B: Survey Instrument

Consent Form
Librarians Who Code: Web and Software Development Skills of Academic Web Librarians in the Southeastern United States

Primary Investigator: Allen Bell (bellma@live.unc.edu)  Research Advisor: Ericka Patillo (patillo@unc.edu)  UNC School of Information and Library Science

Thank you for your interest in this study of the skills, experience, and background of web services librarians and other academic librarians whose job responsibilities include web and/or software development. You have been selected for this survey because your job title suggests that your position may entail such work.

About this study:

What’s involved: This survey consists of 18 multiple-choice questions and two free text questions. It should take 10 to 15 minutes to complete. If you agree to participate in this study, you will be one of approximately 300 people in this research study.

Risks: This survey does not pose any risks to you. Benefits: There are no benefits to your participation in this survey beyond a sense of contributing to the future of your profession in a small way.

Your privacy: By clicking to enter the survey, you are giving permission to use your data in this study. The results of this study will be reported in a master’s paper at the School of Information and Library Science at UNC-Chapel Hill, but the paper will not contain information that will identify you. Your data will be presented anonymously in aggregate statistics. All the information you provide will be used responsibly and will be protected against release to unauthorized persons. Your response will be anonymized by Qualtrics survey software, so that the researchers will only see IP addresses listed rather than any identifying information.

Protection of survey data: Qualtrics’ servers are protected by firewalls and are scanned for vulnerabilities on a regular basis and Transport Layer Security (TLS) encryption is used for the transfer of data.

Payment: You will receive no payment or compensation for participating in this study.

Your rights: You have the right to ask, and have answered, any questions you may have about this research. If you have questions, or concerns, you should contact the researchers listed at the top of this form. All research on human volunteers is reviewed by a committee that works to protect your rights and welfare. If you have questions or concerns about your rights as a research subject you may contact, anonymously if you wish, the Institutional Review Board at 919-966-3113 or by email to IRB_subjects@unc.edu.

Voluntary participation: Your decision whether or not to participate in this study is voluntary. You may choose not to be in the study or to stop being in the study before it is over at any time. If you click on the button below and submit a completed survey, you are indicating your agreement to participate based on reading and understanding this form. If you have any questions, please contact an investigator identified at the top of this form prior to completing the survey.

If you do not wish to participate in this study, please navigate away from this web page. Based on the information above, I agree to participate in this study by clicking the "next" button below.
**Screening question**
Do you use programming languages to develop websites, web applications, or desktop applications in your library job?
- Yes
- No

**Experience**
About how old were you when you first made a web page using HTML/CSS?
- Under 12
- 12-18
- 19-24
- 25-30
- Over 30

About how old were you when you first started learning computer programming, excluding HTML/CSS?
- Under 12
- 12-18
- 19-24
- 25-30
- Over 30

About how many years in total have you worked in a library position that required programming and/or web development skills?
- Less than a year
- 1-3 years
- 4-6 years
- 6-9 years
- 10 or more years

About how many years in total, if any, have you worked in a non-library position that required programming and/or web development skills?
- I have never held such a position
- Less than a year
- 1-3 years
- 4-6 years
- 6-9 years
- 10 or more years
**Education**

Do you hold a Bachelor's degree?
- Yes
- No

In what discipline is your Bachelor's degree?
- Business
- Computer science or information technology
- Education
- Humanities & social sciences
- Math, science, or engineering
- Other

Do you hold a Master of Library and Information Science (MLIS) or similar ALA-accredited degree?
- Yes
- No

In what year did you receive your MLIS degree?
[Free text, restricted to intervals]

Are you currently enrolled in an MLIS program?
- Yes
- No

Please list any other degrees or certifications relevant to web or software development that you hold.
[Free text]

As part of your LIS program, how many courses did you take that were primarily focused on web development, programming, or database management?
- None
- 1-3
- 4-6
- More than 6

Which of the following instructional methods have you used in acquiring your web and/or software development skills? Consider any Computer Science or IT course taken through your LIS program to be an LIS course. Select all that apply.
• Computer science course(s)
• Library/Information science course(s)
• Professional development workshop(s) or course(s) on web development
• Self-taught using instructional books
• Self-taught using websites
• Other (please specify)
  o [Free text]

Rate the usefulness of each instructional method in acquiring your web and/or software development skills. **Consider any Computer Science or IT course taken through your LIS program to be an LIS course.**

<table>
<thead>
<tr>
<th></th>
<th>Not useful</th>
<th>Somewhat useful</th>
<th>Useful</th>
<th>Very useful</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer science course(s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Library/Information science course(s)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Professional development workshop(s) or course(s) on web development</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-taught using instructional books</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-taught using websites</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other (please specify) [Free text]</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Work Environment**

How many work hours did you spend on computer programming and/or web development last week?
• None
• Less than 10
• 10-19
• 20-29
• 30 or more
Which of the following programming / web development skills have you used in the past four weeks for your library job? Select all that apply.

- HTML/CSS
- Database management (e.g. MySQL)
- Server-side scripting (e.g. PHP, Ruby, etc.)
- Client-side scripting (e.g. JavaScript)
- Other scripting languages (Perl, Python)
- Compiled programming languages (e.g. Objective C, Java, C++, etc.)
- Unix command line
- JSON
- XML
- Frameworks (e.g. Django, Rails, Symfony)
- Content management systems (WordPress, Drupal, Joomla, etc.)
- Other (please specify)
  - [Free text]

Which of the following programming / web development skills did you learn through your LIS program? Select all that apply.

- HTML/CSS
- Database management (e.g. MySQL)
- Server-side scripting (e.g. PHP, Ruby, etc.)
- Client-side scripting (e.g. JavaScript)
- Other scripting languages (Perl, Python)
- Compiled programming languages (e.g. Objective C, Java, C++, etc.)
- Unix command line
- JSON
- XML
- Frameworks (e.g. Django, Rails, Symfony)
- Content management systems (WordPress, Drupal, Joomla, etc.)
- Other (please specify)
  - [Free text]

What types of web projects have you worked on in the past 12 months for your library? Select all that apply.

- Library website(s)
- Web app(s)
- Mobile web app(s)
- Mobile app(s) (iOS, Android)
- Desktop app(s)
- Other (please specify)
  - [Free text]
Demographics
What is your age?
[Free text, restricted to intervals]

Which of the following best describes how you think of yourself?
• Female
• Male
• In another way (specify if you wish)
  o [Free text]
• Prefer not to say