Charleston Library Conference

Geek Out: Adding Coding Skills to Your Professional Repertoire

Bohyun Kim
Florida International University, kimb@fiu.edu

Kathryn Harnish
Online Computer Library Center, Inc., harnishk@oclc.org

Bohyun Kim and Kathryn Harnish, "Geek Out: Adding Coding Skills to Your Professional Repertoire" (2012). *Proceedings of the Charleston Library Conference.*
http://dx.doi.org/10.5703/1288284315142

Geek Out: Adding Coding Skills to Your Professional Repertoire[1]

Bohyun Kim, Digital Access Librarian, Florida International University Medical Library

Kathryn Harnish, Senior Product Manager, Online Computer Library Center, Inc.

Introduction

Learning how to program or code has recently become a kind of social phenomenon. For example, Code Year (http://codeyear.com/), a free code-learning website by Codecademy, a company that aimed at teaching the public how to code early this year, attracted as many as 200,000 aspiring programmers within just several days after the site launch.[2] This inspired many professionals such as journalists and marketing and business professionals. Librarians at the American Library Association even formed an official interest group to follow the programming lessons of the Code Year website together.[3]

Librarians' strong interest in programming is no surprise, considering many librarians learn that programming skills are crucial and often essential to making today's library systems and services more user-friendly and efficient for use. Not only for system-customization, programming skills can make it possible to create and provide a completely new type of service by creating a backbone system that didn't exist before or was not offered by any library systems vendor. However, programming skills are not part of most LIS curricula, and librarians often experience difficulty in picking up programming skills. This article will offer some effective strategies to obtain coding skills, cover several common mistakes and obstacles that librarians make and encounter while trying to learn how to program in the library environment, and offer helpful tips and resources. The list of learning and community resources is provided in the appendix.

Why Programming Matters To Librarianship

The information landscape has fundamentally changed since the World Wide Web became the common platform for information and knowledge sharing and storing. Libraries are now expected to not just collect and organize information resources as they have been traditionally doing. As information becomes abundant rather than scarce, the problem that libraries have to tackle has changed from collecting as much as information possible to enabling library users to sift through the vast amount of information, to distinguish what is relevant to a given project from what is not, and to brining overlooked information to users' attention (Kim, Colegrove, & Clark, 2012). On the Web, all of these can be achieved with the right kind of systems and programming functions.

Armed with programming skills, librarians can innovate and revolutionize the way information resources are currently served and used at libraries without relying on a small or often non-existent IT department that is often too overstretched to tend to the needs of a library. Being familiar with the concepts of computer programming alone will enable librarians to communicate better with their IT departments and the library systems vendors. Coding skills can also help librarians to better evaluate many vendor products and to customize and improve the configurable parts of the third-party library systems. In addition, librarians will find their coding skills useful for streamlining workflows by automating repeated work routines. Lastly, librarians can build something new and create new library services with their programming skills either on their own or in collaboration with programmers in their organization.

How to Obtain Coding Skills, Effectively

1. Pick a language and concentrate on it.

There are a huge number of resources available on the Web for those who want to learn how to program. Often librarians start with some knowledge in markup languages such as HTML and CSS. These markup languages determine how a block of text are marked up and presented on the computer screen. On the other hand,

http://dx.doi.org/10.5703/1288284315142

programming languages involve programming logic and functions. An understanding of the basic programming concepts and logic can be obtained by learning any programming language. There are many options, and some popular choices are JavaScript, PHP, Python, Ruby, Perl, etc., but there are many more. For example, if you are interested in automating tasks in Microsoft applications such as Excel, you may want to work with Visual Basic. If you are unsure about which language to pick, search for a few online tutorials for a few languages to see what their different syntaxes and examples are like. Even if you do not understand the content completely, this will help you to pick the language to learn first.

2. Write and run the code.

Once you choose a language to learn, there are many paths that you can follow. Taking classes at a local community college or through an online school may speed up the initial process of learning, but it could be time-consuming and costly. Following online tutorials and trying each example is a good alternative that many people take. You may also pick up a few books along the way to supplement the tutorials and use them for reference purposes.

If you decide on self-study, make sure that you actually write and run the code in the examples as you follow along the books and the tutorials. Most of the examples will appear simple and straightforward. But there is a big difference between reading through a code example and being actually able to write the code on your own and to run it successfully. If you read through programming tutorials and books without actually doing the hands-on examples on your own, you won’t get much benefit out of your investment. Programming is a hands-on skill as much as an intellectual understanding.

3. Continue to think about how coding can be applied to your library.

Also important is to continue to think about how your knowledge can be applied to your library systems and environment, which is often the source of the initial motivation for many librarians who decide to learn how to program. The best way to learn how to program is to program, and the more you program the better you will become at programming. So at every chance of building something with the new programming language that you are learning, no matter how small it is, build it and test out the code to see if it works the way you intended.

4. Get used to debugging.

While many who struggle with learning how to code cite lack of time as a reason, the real cause is likely to be failing to keep up the initial interest and persist in what you decided to learn. Learning how to code can be exciting, but it can also be a huge time-sink and the biggest source of frustration from time to time. Since the computer code is written for a machine to read, not for a human being, one typo or a missing semicolon can make the program non-functional. Finding out and correcting this type of error can be time-consuming and demoralizing. But learning how to debug is half of programming. So don’t be discouraged.

5. Find a community for social learning and support.

Having someone to talk to about coding problems while you are learning can be a great help. Sign up for listservs where coding librarians or library coders frequent, such as code4lib and web4lib to get feedback when you need. Research the cause of the problem that you encounter as much as possible on your own. When you still are unsure about how to go about tackling it, post your question to the sites such as Stack Overflow for suggestions and answers from more experienced programmers. It is also a good idea to organize a study group with like-minded people and get support for both coding-related and learning-related problems. You may also find local meet-ups available in your area using sites like MeetUp.com.

Don’t be intimidated by those who seem to know much more than you in those groups (as you know much more about libraries than they do, and you have things to contribute as well), but be aware of the cultural differences between the developer community and the librarian community. Unlike the librarian community that is highly accommodating for new librarians and sometimes not-well-thought-out questions, the developer community that you get to interact with may appear much less accommodating, less friendly, and less patient. However, remember that reading many lines of code, understanding what they are supposed to do, and helping someone to solve a problem occurring in those lines can be time-consuming and difficult even to a professional programmer. So it is polite to do a thorough research on the Web and with some reference resources first before asking for others' help. Also, always post back a working solution when your problem is solved, and make sure to say thank you to people who helped you. This way, you are contributing back to the community.

6. Start working on a real-life problem "now."

Librarians are often motivated to learn how to code in order to solve real-life problems they encounter at their workplace. Solving a real-life problem with programming is therefore the most effective way to learn and to keep up the interest in programming. One of the greatest mistakes in learning programming is putting off writing one's own code and waiting to work on a real-life problem for the reason that one doesn't know yet enough to do so. While it is easy to think that once you learn a bit more, it would be easier to approach a problem, this is actually a counter-productive learning strategy as far as programming is concerned, because often the only way to find out what to learn is by trying to solve a problem.

7. Build on what you learned.

Another mistake to avoid in learning how to program is failing to build on what one has learned. Having solved one set of problem doesn't mean that you will remember that programming solution you created next time when you have to solve a similar problem. Repeating what one has succeeded at and expanding on that knowledge will lead to a stronger foundation for more advanced programming knowledge. Also instead of trying to learn more than one programming language (e.g., Python, PHP, Ruby, etc.) and/or a web framework (e.g., Django, cakePHP, Ruby On Rails, etc.) at the same time, first try to become reasonably good at one. This will make it much easier to pick up another language later in the future.

8. Code regularly and be persistent.

It is important to understand that learning how to program and becoming good at it will take time. Regular coding practice is the only way to get there. Solving a problem is a good way to learn, but doing so on a regular basis as often as possible is the best way to make what you learned stick and stay in your head.

While is it easy to say practice coding regularly and try to apply it as much as possible to the library environment, actually doing so is quite difficult. There are not many well-established communities for fledgling coders in libraries that provide needed guidance and support. And while you may want to work with library systems at your workplace right away, your lack of experience may prove problematic in gaining a necessary permission to tinker with them. Also as a full-time librarian, programming is likely to be thrown to the bottom of your to-do list.

Be aware of these obstacles and try to find a way to overcome them as you go. Set small goals and use them as milestones. Be persistent and don't be discouraged by poor documentation, syntax errors, and failures. With consistent practice and continuous learning, programming can surely be learned.

Support for Librarian Coders from Libraries

While libraries often lament the lack of programmers on their staff, libraries have been quite passive in supporting their staff who are interested in learning how to program and in applying the knowledge to the library systems and services (Kim, 2011). Libraries can, however, play a significant role in creating and fostering an environment in which more librarians are encouraged to become familiar with programming concepts and to obtain a certain level of coding skills that will be useful to troubleshooting and simple scripting.

Particularly if you are a manager, you can facilitate this process by identifying interested staff members and providing some research and development time aside from their regular work. Ask the staff to present what they have learned and coded, and encourage them to think about how it can be applied to the library services and/or systems. Offer opportunities for them to participate in the real coding project at the library. Since these people are just starting to learn, it would be also good to offer guidance by more experienced programmers if possible. Joint projects with the library IT department and/or the IT department outside the library can be a good way to create the synergy between librarian coders and programmers who work for libraries. This way, a library can connect the staff's interest in programming with the tangible improvement of the library services and systems.

Many library managers appreciate the importance of programming skills, but they often fail to realize that they can help their own library staff to obtain those skills. Supporting the library staff who are interested in learning how to code does not mean that the library will stop hiring programmers from the outside of the library. Nor does it mean that all those staff members who learn how to program will obtain a very high level of programming skills. However, by encouraging more librarians to become familiar with programming concepts and practices, libraries will be taking one step closer to successfully adjusting themselves to the changed information environment in which the access to and the use of information and knowledge is no longer separable from the technology that is used to create, distribute, deliver, and preserve such information and knowledge.

Platform U by OCLC

Platform U is a series of coding courses sponsored and currently in development by OCLC. It will address a range of skill levels with different classes starting with a five-week “Learn to Code” class and then move to more advanced programs. The goal of this new learning program is to provide learning opportunities and support for library staff with an interest in coding. Platform U can be used as one of many resources and tools for those who are interested in learning how to code in the context of libraries. More details will be announced by OCLC when the program is unveiled in the near future.

References

Kim, B. (2011, February 21). Why not grow coders from the inside of libraries? Retrieved from http://www.bohyunkim.net/blog/archives/1099

Kim, B. (2012, December 10). Effectively learning how to code: Tips and resources. Retrieved from http://acrl.ala.org/techconnect/?p=2460

Kim, B., Colegrove, P., & Clark, J. (2012). I can do it all by myself: Exploring new roles for libraries and mediating technologies in addressing the do-it-yourself mindset of library patrons. [PowerPoint slides]. Retrieved from http://www.slideshare.net/bohyunkim/i-can-do-it-all-by-mysef-exploring-new-roles-for-libraries-and-mediating-technologies-in-addressing-the-diy-mindset-of-library-patrons

Kim, B., & Harnish, K. (2012). Geek out: Adding coding skills to your professional repertoire. [PowerPoint slides]. Retrieved from http://www.slideshare.net/bohyunkim/geek-out-adding-coding-skills-to-your-professional-repertoire

Segall, L. (2012, January 6). Code Year draws 200,000 aspiring programmers. *CNN Money.* Retrieved from http://money.cnn.com/2012/01/06/technology/code_year/index.htm

Appendix

A. Resources for learning

- Codecademy http://www.codecademy.com/
- Coursera https://www.coursera.org/category/cs-programming (This site offers many programming courses.)
- Github https://github.com/
- How to Design Programs (2nd ed.) http://www.ccs.neu.edu/home/matthias/HtDP2e/
- Mozilla Development Network https://developer.mozilla.org/en-US/
- Online tutorials (e.g., W3C tutorials http://www.w3schools.com/w3c/default.asp but there are many others)
- Programming books in the Head First and the Visual QuickStart series.
- Recommendations from Code4Lib http://wiki.code4lib.org/index.php/One_recommended_tool/resource_for_n00bs
- Safari Books Online http://my.safaribooksonline.com/ (A subscription service for programming books.)
- Stack Overflow http://stackoverflow.com/

B. Communities

- Catcode wiki /IRC (#catcode on Freenode): http://catcode.pbworks.com/w/page/49328692/Welcome%20to%20CatCode
- Code4Lib listserv / Wiki/ IRC (#code4lib on Freenode): https://listserv.nd.edu/cgi-bin/wa?A0=code4lib (Also: Web4Lib, XML4Lib, Sys4Lib, Foss4Lib, Sec4LIB, etc.)
- Devchix: http://www.devchix.com
- Library Code Year IG (ALCTS/ LITA): http://connect.ala.org/node/167971
- LibTechWomen: Twitter (@LibTechWomen) IRC (#libtechwomen on Freenode)
- Local coding community meet-ups: PHP, Python, Ruby, etc. (Try searching the MeetUp.com website.

[1] This paper is based upon the talk given at Charleston Conference XXXII, November 9, 2012. The presentation slides are available at SlideShare.net. See Kim & Harnish (2012). Part of this article has been also published in ACRL TechConnect blog. See Kim (2012).

[2] Segall, L. (2012, January 6). Code Year draws 200,000 aspiring programmers. *CNN Money*. Retrieved from http://money.cnn.com/2012/01/06/technology/code_year/index.htm

[3] Library Code Year Interest Group. Retrieved from http://connect.ala.org/node/167971