Programming for Librarians: Where to begin

Péter Király Peter Verhaar

2024-06-04

This guide aims to provide library professionals with some guidance on how and where to begin their journey into learning a programming language.

Introduction

Many of the tasks taking place within libraries encompass activities which can be automated. Librarians may need to do bulk downloads of resources from the web, adjust the values in a database or manipulate images. Instead of doing such tasks manually, it is evidently more efficient to develop computer code which can perform such clerical tasks repeatedly. It can be very helpful for librarians, because of this, to become proficient in a programming language. Examples of programming languages include Python, R, Java, Perl or PHP.

Programming can be described as the process of using computers to address specific problems. Computer programs essentially consist of a sequence of statements which explain how an activity can be automated. Programming languages generally implement a certain algorithm. An algorithm is an explicit and unambiguous description of the steps that need to be followed to arrive at a well-defined end result. The algorithm constitutes the logic of the program. The statements or commands in a programming language also need to be expressed in a format that computers can understand. They need to be expressed using a syntax, which prescribes how words, numbers of punctuation marks ought to be used

There are a number of different languages in which you can write computer code. While it is difficult to find reliable data about this, it feels safe to claim that Python and R are currently the most widely used languages in the library sector. When you start learning how to program, it is probably best to focus on one language initially and to develop skills in one specific programming language. It must be stressed, however, that when you develop your expertise in one concrete language, this will inevitably help you to improve your computational thinking and to broaden your understanding of how programming languages function in general. Once you have learned about the nuts and bolts of one language, it becomes much easier to gain mastery over another language.

Which programming language should you focus on? Python vs R?

Python, firstly, is a free and open source, general purpose language. It is widely used to carry out data science tasks, such as cleaning, enriching, analysing and visualising data. Python makes use of a sparsee, readable and intuitive syntax, and this makes the language relatively easy to learn for beginners. Within the Python community, numerous code libraries have been shared which extend the basic functionalities of the language.

R, is another free and open source programming language that was created originally with a specific focus: to support statistical analyses. Later it was developed towards a more general direction with the help of thousands of software packages created by the R community and today it is used to solve as many types of problems as Python or Java. R's syntax and its most important data structures are quite different from that of Python, and it requires a different way of thinking.

Because of this strong data analysis background we suggest you choose R if you would like to run core data analysis tasks that require a more advanced statistical toolbox, and does not require many additional steps before and after the calculations. Python on the other hand provides somewhat less statistical functionalities, but supports much more general purpose tasks out of the box like file management, string, date and time manipulation, computer networks etc. None of these tasks are impossible with the other programming languages, but their conveniences are different. In data science related tasks they are quite close, e.g. Python's Pandas library and R's data frames (particularly in the Tidyverse package) have similar functionalities. Another - subjective - factor of the decision could be the aesthetics of the graphics Python and R produces: both of them have more than one plotting libraries, each are easy to recognise by their distinctive graphical style.

Relevance to the Library Sector (Case Studies/Use Cases)

Programming languages can be used for a variety of library tasks.

- A common application is to convert one data format into another data format. A computer system may export descriptions in the MARCXML format, while another application may demand JSON input. Such transformations can be carried out using a computer program.
- It is also possible to download large numbers of files using computer code.
- Programming languages can also be used to analyse and to visualise data in a spreadsheet or in a CSV format. As such, they may help others to interpret the data or to see patterns or regularities in large datasets.
- Applications based on AI and Machine Learning are typically created using a programming language.

The British Library, The National Archives and Birkbeck University in the UK partnered on a trial of a one-year part-time postgraduate Certificate (PGCert), Computing for Cultural Heritage, as part of a £4.8 million University skills drive in 2019-2021. The 20 staff projects devised and undertaken by the students demonstrate the wide range of what programming skills can enable for those working in a library setting.

The code4lib journal publishes many articles with case studies on applications in library contexts.

Hands-on activity and other self-guided tutorial(s)

To develop a basic understanding of the main features of Python, you can follow the Introductory course on Python that was developed at the University of Leiden.

It is a self-paced course which you can follow to familiarise yourself with central concepts such as variables, operators, flow control, data structures and functions. The tutorial also explains some of the code libraries which can be used more specifically for data science tasks such as data acquisition, working with APIs, data analysis and data visualisation. The tutorial includes a large number of exercises with which you hone your programming skills, together with model answers which can help you if you get stuck.

Once you gained a basic proficiency of the central Python concepts, you can follow the tutorial "Python for Librarians", which was developed by the Library Carpentry.

To get started with R we recommend the Library Carpenty's tutorial "Introduction to R" which teaches you how to do basic data science tasks on tabular data, such as spreadsheets or relational database tables.

There are a number of useful resources to improve your programming skills specifically with library related materials.

- Tim Sherratt: GLAM Workbench (2021-). As its introduction says: "The GLAM Workbench is a collection of Jupyter notebooks to help you explore and use data from GLAM institutions (that's galleries, libraries, archives, and museums). It includes tools, tutorials, examples, hacks, and even some pre-harvested datasets." Sherratt focuses on Australian and New Zealand datasets, but you can reuse the code for other sources. The notebooks are written in Python. Jupyter Notebook, a web-based interactive computing platform that combines live code, equations, narrative text, visualisations, interactive dashboards and other media. You can find many tutorials and GLAM related scripts in this form.
- Programming Historian is a set of programming tutorials (both in Python and R) focusing on particular computational problems (such as network analysis, data managements, APIs, machine learning) made mainly for historians, but several tutorials use GLAM data.

Recommended Reading/Viewing

Paul Vierthaler's Hacking the Humanities Tutorials video series is quite useful for beginners.

We can highly recommend this recent book *Hands-On Data Science for Librarians* written by Sarah Lin and Dorris Scot (Boca Raton : CRC Press, 2023, ISBN 978-1-032-10999-2).

Though not written specifically for librarians, but probably the best introduction to the topic of data science with R is *R for Data Science*. 2nd edition by Hadley Wickham, Mine Çetinkaya-Rundel and Garrett Grolemund (O'Reilly, 2023, ISBN 978–1-492-09740-2), and available online.

Finding Communities of Practice

When you're first getting started in programming, being able to ask others questions when you get stuck is so important! Code4lib is a volunteer-driven collective of folks that has been around since 2003 and amongst their many resources are a large range of chat rooms for folks interested in the convergence of technology and cultural heritage. Visit Chat | Code4Lib to find the discussion you need or join their mailing list.

You might also want to have a look on Github, which is one of the largest source code repositories on the web, for the "code4lib" tag to denote repositories that are somewhat relevant for the library community. They are not tutorials, but during your learning process you can check if there is a software solution there already for your particular problem, and you can even try to engage with one or more tools as a contributor. Contributions are always welcome on #code4lib projects, and even if you have just begun programming you can provide important feedback about the usage of a tool, or you can improve documentation as a first step. See our page on using GitHub for more on that!