

Gfmi
GLOBAL FINANCIAL MARKETS INSTITUTE

Article

2019

Practical FinTech for Financial Professionals

by Cara Marshall, Ph.D.

Practical FinTech for Financial Professionals

FinTech Defined

The recent growth of financial technology, or “fintech,” affects us in many ways. Upon hearing the term “fintech,” you may envision technology related to cryptocurrency, mobile payments, money transfers, crowdfunding, robo-advisors, and asset management. In its broadest definition, fintech is “technologies used and applied in the financial services sector, chiefly used by financial institutions themselves on the back end of their businesses” (Marr, 2017).

Fintech may be visible to the consumer, as in the development of mobile applications for personal banking which allows users to check balances or deposit checks via the Internet. However, fintech is more often invisible to the consumer, and simply works to improve productivity and streamline data processes on the back end of a financial firm.

In this article, we evaluate the evolution of technology in the financial industry, discuss some core programming concepts, survey the commonly used programming languages used by financial professionals in the industry, and discuss the practical ways that computer programming can dramatically increase productivity for many financial professionals.

Learning to code is increasingly becoming an essential part of most graduate programs in finance. Many undergraduate students studying business take an introductory level computer programming class. The STEM (science, technology, engineering, and mathematics) initiative in our primary and secondary schools is well underway, and several companies offer summer “coding” camps targeting our youth as well. Learning the basics of coding is advantageous in today’s society.

Evolution of Technology in the Finance Sector

Beginning in the 1950s with credit cards and later ATMs, followed by electronic trading, mainframe computers, and later the Internet, “these five decades of developments have created a financial technology infrastructure which most people never think about, but use almost every day. It’s also important to note that throughout that 50-year period, fintech developments were also creating more sophisticated risk management, trade processing, treasury management and data analysis tools at the institutional level for banks and financial services firms. While these systems are not apparent to retail banking customers, they make up a multibillion industry aimed at supporting the needs of the financial services sector” (Desai, 2015).

The finance industry has become much more technology-driven and many processes have become automated. Julien Courbe of PricewaterhouseCoopers writes, “Financial institutions have a lot on their plate: emerging competitors, shifting demographics, rising customer expectations and changing regulations. Technology offers solutions, allowing financial institutions to cut costs and become more efficient at what they do” (Courbe, 2016). As the industry continues to move in the direction of automation, finance professionals with programming knowledge and technical skills are, and will continue to be, in high demand.

Basic Programming Concepts

Regardless of the specific programming language you intend to use, there are several core concepts that must be understood to start coding. These are the fundamentals that, once grasped, allow you to move from one language to another relatively easily. As technology evolves and new computer languages are developed, you must learn new syntax, but in general the basic concepts remain the same.

Code is written inside an Integrated Developer Environment (IDE), which is essentially a text editor that understands certain key words associated with the programming language. Different languages have preferred IDEs, and there may be more than one IDE for a given language. While code can be written inside a plain text editor such as Notepad, it is helpful to use the associated IDE so that reserved words are recognized and often color coded. In certain cases, the IDE may contain “Intellisense,” which can make suggestions for revisions or additions to code. IDEs often have debugging tools that can be very helpful as well.

One fundamental concept is the ability to define (or declare) variables and set variable types. Variables are storage containers that will hold information inside the program. That information can be manipulated by the code and later output when necessary. The type of information that will be stored, can be specified (variable types include: integers, numbers with decimals, dates, strings of text, certain objects, etc.).

One needs to understand how to get input from the user and how to output results. There may be several ways to input/output. One can build functions and use mathematical operators within the code. Logic is routinely used inside code as well. Logic is used not only to determine an output, but also to build workarounds for preventing errors that may result in the program crashing.

Perhaps the most powerful concept is looping. Looping is used to repeat your code multiple times. For example, imagine that you wish to perform the same analysis on many spreadsheets that are structured the same way. You can write code to loop through all files in a particular folder and perform that same analysis on every single file. Different looping techniques can be utilized depending on the application required.

Learning these basic computer programming concepts, regardless of the programming language, gives you the backbone required to get started. Once you have an understanding of how to apply these concepts, it is relatively easy to learn a new programming language.

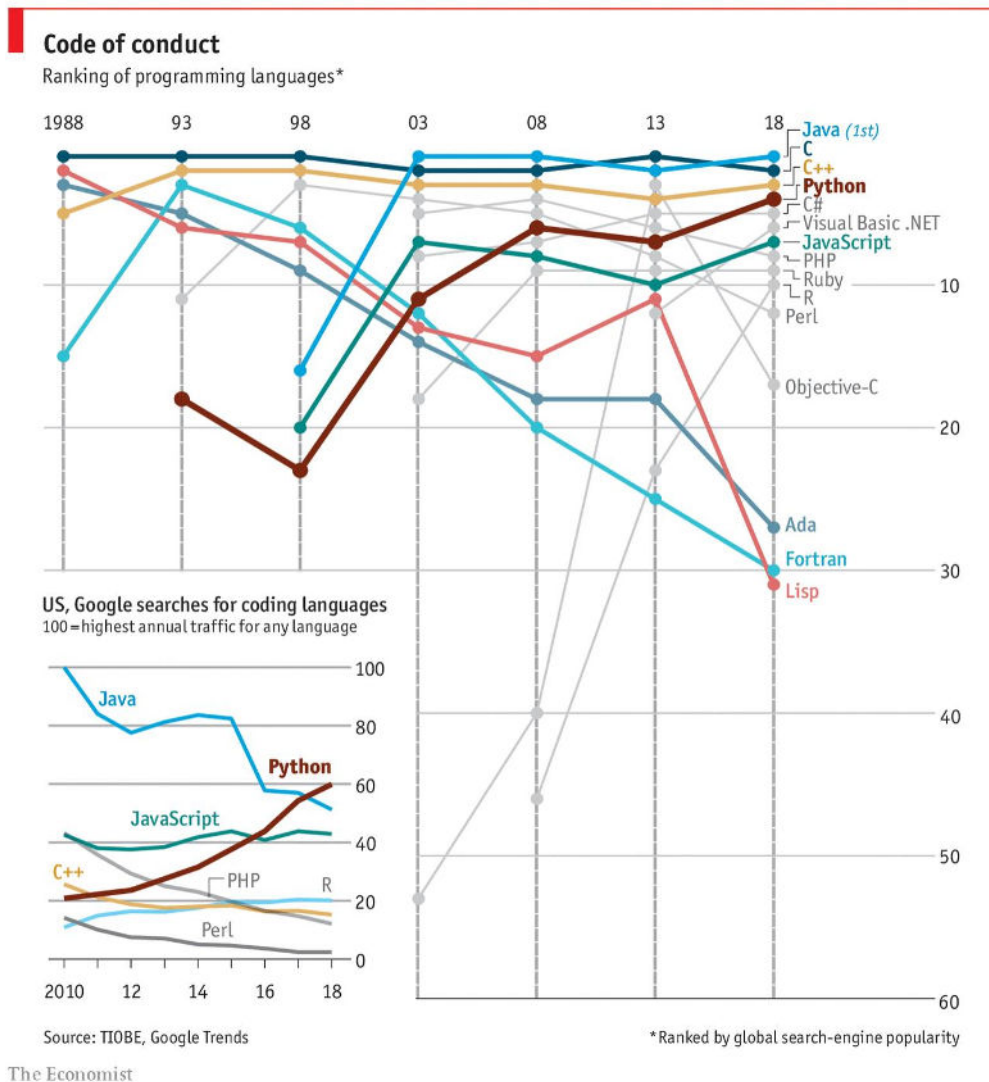
Popular Programming Languages for Financial Professionals

Those analysts working with statistics would find it helpful to understand coding in sophisticated statistical software packages like SAS, STATA, SPSS, or R. These programs allow the user to run regression analysis to predict trends and evaluate relationships between variables such as macroeconomic indicators, asset returns, sales, production, and the like. MATLAB is another specialized software tool for quantitative analysis requiring matrix manipulation. MATLAB is useful for transforming calculus and linear algebra, which is typically very theoretical, into practical application.

Structured Query Language (SQL) is another useful language for financial professionals. SQL is used to work with databases and SQL is the underlying language behind Microsoft Access. This language is helpful when working with large datasets and merging data from several tables or querying subsets of data. SQL can also be utilized inside other programming languages.

Apart from these specialized mathematical and database tools, coding is quite useful for a variety of roles in finance. In particular, Visual Basic for Applications (VBA), the underlying language behind the Microsoft Office Suite, is an extremely useful language for anyone who works with Microsoft Excel (and really, who doesn't?). VBA can be used to automate routine tasks. For example, let's say you get a file containing your desk's trade positions every week, and you have to regularly parse and clean the data, then perform some VLOOKUPs and calculations on the data, before finally creating a pivot table and sending it off to your manager. It may take several hours to perform this same set of tasks that you perform every week. VBA can be used to create a subroutine (macro) that will perform these actions on any file you pull up. Once the code is written, you simply run the macro (which can even be assigned to a keyboard shortcut), and it will take the computer merely a few seconds to perform that series of tasks from start to finish, which once took you several hours.

C, C++, and Java are common, general purpose programming languages used by developers for all sorts of applications. Prior to the introduction of Python, an open-source programming language, these languages were being used by quants in the finance industry to develop algorithms. See the graph below depicting popularity of various programming languages over the span of thirty years from 1988 through 2018.



Python has evolved, and grown quite rapidly as depicted in the chart, as a wonderful tool for working with large datasets (“big data”), performing analysis, and visualizing the data graphically, among other things. Python shares the same capabilities of languages that preceded it, however, writing, compiling, and debugging the code has become more user-friendly. In addition, because Python is open-source, programmers have written and shared many libraries (also called modules or packages), that enable us to quickly develop sophisticated code without having to reinvent the wheel.

Python’s “avid practitioners,” known as Pythonistas, have uploaded 145,000 custom-built software packages to an online repository. These cover everything from game development to astronomy, and can be installed and inserted into a Python program in a matter of seconds. This versatility means that the Central Intelligence Agency has used it for hacking, Google for crawling webpages, Pixar for producing movies and Spotify for recommending songs. Some of the most

popular packages harness ‘machine learning’, by crunching large quantities of data to pick out patterns that would otherwise be imperceptible” (Kambayashi, 2018).

Popular libraries frequently used by financial professionals include: Math, NumPy, SciPi, Pandas, and Matplotlib. Math contains some basic mathematical functions (natural log, square root, etc.). NumPy and SciPi focus on numerical programming specializing in working with matrices. Pandas contains tools related to data analytics. Matplotlib allows you to generate charts, histograms, and the like so that you can visualize your data. There are also many very specialized libraries that will perform backtesting, option pricing, analyze the markets, perform algorithmic trading, run technical analysis, pull in financial data from the web, and the like. These libraries can be imported and utilized directly in your Python code.

Popular Coding Applications for Financial Professionals

Knowledge of programming benefits financial analysts, portfolio analysts, quantitative analysts, risk managers, compliance officers, auditors, and those in similar roles. Anyone who works with large financial datasets or complex models will be in high demand with programming skills and technical knowledge. Coding can be used to automate processes that get performed regularly, and then perform these tasks repeatedly.

Programming helps us to manage, process, and manipulate data. Creating code with loops allows us to perform analysis over and over on many sets of data, whether the data is in one spreadsheet, across many spreadsheets, or across many files. Code can also be written to send emails in Outlook automatically, scrape data from a website to build your own data pipeline, build your own Excel functions, develop Excel Add-ins, perform backtesting on a strategy, run Monte Carlo simulations, and much more.

Conclusion

Students across the globe are now being trained on basic coding concepts as part of their academic curriculum. For financial professionals, the ability to write code can transform your life in a way that makes you much more productive. In particular, VBA and Python are extremely useful languages for those financial professionals working with Microsoft Excel or other types of large datasets. Fortunately, the concepts gained from an introduction to programming in any language can easily be transferred to other programming languages as they are developed, as “change is the only constant in life.”

References

1. Courbe, Julien, "Financial Services Technology 2020 and Beyond: Embracing disruption," <https://www.pwc.com/gx/en/financial-services/assets/pdf/technology2020-and-beyond.pdf>
2. Desai, Falguni, "The Evolution of FinTech," Forbes, 13 Dec 2015, <https://www.forbes.com/sites/falgunidesai/2015/12/13/the-evolution-of-fintech/#4ff78d817175>
3. Kambayashi, Satoshi, "Python has brought computer programming to a vast new audience," The Economist, 19, Jul, 2018, <https://www.economist.com/science-and-technology/2018/07/19/python-has-brought-computer-programming-to-a-vast-new-audience>
4. Marr, Bernard, "The Complete Beginner's Guide To FinTech Everyone Can Understand," Forbes, 10 Feb, 2017, <https://www.forbes.com/sites/bernardmarr/2017/02/10/a-complete-beginners-guide-to-fintech-in-2017/#5c8611833402>

About the Author: Cara Marshall, Ph.D.



Cara Marshall has the unique background of the academics (graduate and undergraduate levels) combined with a financial markets career. With over fifteen years of classroom experience, she knows the art of designing, developing and delivering an engaging learning program, keeping the students actively participating in the process. She also has financial industry experience, starting as an intern with Merrill Lynch, and working her way up to an independent consultant to some of the largest investment banks, federal agencies and hedge funds and other banks for the past sixteen years. In addition, she is a published author, writing journal articles and a book that focus on her areas of special interest, such as financial engineering; derivatives; pricing and risk management, amongst others. This dual-experience background – of the markets as well as the academics – enables Cara to apply hands-on practical experience and in-depth knowledge to beginners as well as to more advanced students.

Cara's specific topics of expertise include:

- Microsoft Excel for Financial Professionals (Introductory/Intermediate/Advanced)
- Programming in Visual Basic for Applications (VBA) in Excel
- Programming with Python
- Financial Modeling



Copyright © 2019 by Global Financial Markets Institute, Inc.
PO Box 388
Jericho, NY 11753-0388
+1 516 935 0923
www.GFMI.com