

# How I Learnt Python in 15 Days

Vikas Dhongde, Accenture Solutions Pvt. Ltd., Mumbai, India

## ABSTRACT

With the advent of Big Data and Analytics few years back, programming languages supporting such number crunching, data transformation and visualization gained much of their stardom that they enjoy today. That's when erstwhile dormant (as I was too obsessed with SAS® yet curious programmer in me woke up to take on these new challenges. Naturally first choice was Python as it being a programmer's language as opposed to R. As a working professional and minimal time to spare during weekdays, I set an ambitious target of 15 days including weekends. With sheer zeal for learning something new and with programming acumen gained over years, I embarked upon a journey to see this new language and to check how it fares against my favorite i.e. SAS.

## INTRODUCTION

Speaking of Python, are you curious about today's buzz words in our industry? is learning Python on your to do list? do you want to learn Python on your own? or are you a SAS programmer and just want to add another skill to your holster?

If answer to any of those questions is Yes, but you are not sure how to begin or what approach to take or if you can learn it by yourself at all...fear you not. Some time back I was in the same place as you and as I sit down to write this paper, I remember how confused I was. Nonetheless I can assure you that learning Python (or anything for that matter) on your own is not as daunting of a task as it may seem. All you need is an eager mind, clear approach, unwavering dedication and in case of Python: sound logic!

This paper assumes that reader has fair amount prior programming experience which would act a foundation to achieve this mammoth task of learning Python *in 15 Days*. At the outset let me state that this paper does not intend to teach you Python or explain any of its codes or functionality. It presents my experience of learning it, along with my strategy and conclusions with the hope that it would help budding programmers in their pursuit of similar goals. It can work for you as a roadmap to become a Python programmer in life science industry. It also is a tale of a SAS programmer who metamorphosed into a working Python programmer in 15 days.

## BACKGROUND

We live in a highly data-driven and technology-oriented world which is now testing the waters of Artificial Intelligence. A common contemporary cliché puts it aptly: Data is the new currency. In just past two decades we saw leaps and bounds increase in volumes and variety of data. There is ever growing need for tools and personal that can support not only data processing and analysis but deriving insights, making decisions and predicting accurately.

Our own industry is not immune to these developments, with things like wearable techs, e-diary, development of genetic databases and talk of virtual clinical trials. Hitherto, though SAS is undisputed leader in pharmaceutical industry, some inroads have been made by tools like R and to some extent Python, however when it comes to the fields of Analytics, Machine Learning and Artificial Intelligence Python is way ahead.

That's when ever-eager, learning-happy programmer in me decided to take on this trending skill. 15 days thing was result of an idea to learn most essential elements of the language as soon as possible and then jump into the real-world Python Programming.

## WHY PYTHON?

Python is extremely versatile high-level, general purpose programming language (GPL) which has widespread applications ranging from websites, applications and games development to Data Analytics, Machine Learning and Artificial Intelligence. There were many reasons to choose Python but some that really appealed to me are:

- a) A programmer's language- If I was a statistician, I might have chosen R but programmer at heart, it was always going to be Python
- b) Simplicity and readability- Although most flaunted, this is arguably Python's strongest suit; and you must see it to believe it
- c) Strong OOP, robust libraries and community support- Python draws its power from strong Object-Oriented Programming (OOP) features like polymorphism, robust packages and healthy community support
- d) Free- Yes! Its free
- e) Rising popularity- Indeed, it was Python's popularity that propelled me towards this undertaking

## HOW CAN THIS HELP YOU?

Now, if you are wondering if you can emulate something similar, it would be good to know what kind of a programmer you are.

- a) Seasoned programmer with multiple years of experience on general purpose programming languages:
  - It's easiest for you, you can pick my strategy as is or tweak it as you want
- b) Seasoned SAS programmer with multiple years of experience on SAS programming:
  - It's still easy although not as easy as the first ones. You may have to change your outlook towards few things, for ex. GPL nature of Python against more of an ETL functionality of SAS
- c) A beginner to the field of programming:
  - If you're one of these, let me be honest and say this is NOT for you. Don't get me wrong, am not saying you can't learn Python; all I am saying is don't try to do it in 15 days. Spend some more time on it and go for in depth learning experience. Having said that you can still use the plan I put forward and tweak the timelines per your convenience.

As for me, I belong to second category, who loves programming and this my journey and experience of learning a whole new language

## EGG: "THE FOUNDATION"

Obviously, first thing to do was to get all the requisites in place and make sure I am well prepared:

- a) **Define specific goal**

As a hardcore SAS programmer from life sciences industry, it's only natural that my goal was clear: "I wanted to learn it as an alternative to SAS, use this learning as foundation and build on it from Data science perspective".

This is particularly important as it helps in narrowing your focus and prevents you from wandering off in vastness of Python.
- b) **Get a good reference book**

Having a good reference book by your side when learning is paramount, not to say that you can't do it without it, but it will provide more structured and linear approach. As I said earlier, I wanted a quick know-how of the language; so, I referred 'A Whirlwind Tour of Python' by Jake VanderPlas from O'Reilly publications. It's a good book for experienced programmers who want to quickly learn most essential parts of the language. 'Learn Python the Hard Way' by Zed Shaw is yet another good book if you want a DIY approach
- c) **Finalize online reference sources, practice platforms**

I also referred Python official documentation sometimes when I needed more details about certain thing. Watching relevant YouTube videos can be a great way to revise when you're not actively learning. There are some online practice platforms (like hackerrank) which provide incremental exercises, this is great way of getting your hands dirty with Python.
- d) **Get Python installed and running**

Being open source, this was easy. You can download it from downloads section of Python.org. For scripting, although many programmers use third party text editors, I prefer IDLE that comes bundled with stock Python for its simplicity.
- e) **Set up a detailed plan for next 15 days**

Next thing to do was devise a 15-day detailed learning plan along with time allocation. It was to act as a motivator and guide through my journey. Having a clearly defined goal will help you come up with relevant and concise learning plan.
- f) **Programming acumen**

And finally, my prior programming experience. Something that I didn't have to exclusively prepare but without which this mammoth task wouldn't have been possible. It was to be my key enabler

## CATERPILLER & PUPA: "THE FEEDING & THE TRANSITION"

Now that I had completed the groundwork, it was time to start actual learning.

- I started with Python environment and GUI. Although the concept of interpreter and separate IDLE/text editor was alien to me as a SAS programmer, I liked the experience of it. I loved the speed of interpreter and similarity of IDLE with SAS enhanced editor. I missed verbose SAS log but liked short and sweet Python exceptions!
- Language elements were really a piece of cake as most of the concepts were not much different than SAS or other programming languages I knew. I was able to cover most of it in less than a week and even save some time for more complex topics. All that programming experience did pay rich dividends indeed!
- Special mention: WHITESPACES and case-sensitivity, they need some real getting used to. As SAS is not case sensitive except when it comes to strings, and it's also indifferent to spaces and indentations between statements or code blocks. In Python it makes a big difference. Practice is your only salvation.
- From there it was bit uphill but still nothing too difficult. strong Object-Oriented Programming (OOP) was first serious thing for me as I was not so clear with that concept. I spent around three days reading and learning in depth about it. If you want to develop complex applications in future, it's imperative that you have clear understanding of this topic.

- Once I covered the language elements, I started with Data science specific packages. I was really looking forward to this, Afterall these are the things most extensively used in real world Python programming for data sciences. Learning NumPy, Pandas and Matplotlib was particularly fun as I could explore ETL functions of Python, try and emulate my routine SAS work with Python scripts.
- It was becoming evident that for any given purpose compared with SAS, Python code is much smaller, easier to read and therefore debug and would most probably be similar even if two different programmers wrote it.
- Concurrent to the learning, I was extensively practicing things on the go. I was able to write complex medium size scripts. Even though I was making lot of mistakes I thought I was learning because of them.
- I could feel my transformation from the Egg to the Caterpillar to the Pupa.

Here is My learning plan for your reference (Please note that contents listed here are only representative, and not comprehensive):

Timeline	Topics	Contents
Day 1-3	<ul style="list-style-type: none"> <li>• Environment, Basics</li> <li>• Data types &amp; structures</li> <li>• Python files</li> <li>• Libraries/Packages/Modules</li> </ul>	Python GUI, Environment Semantics, Keywords Data Types, Variables & Operators Objects, Lists, Tuple, Set, Dictionaries Built-in modules and packages
Day 4-6	<ul style="list-style-type: none"> <li>• Strings &amp; Statements</li> <li>• Control Flow</li> <li>• Conditional processing</li> <li>• Iterative processing</li> </ul>	Assignments, Expressions Indexing, Slicing and Comprehensions if, else, elif, for, while, break, continue, pass
Day 7-9	<ul style="list-style-type: none"> <li>• Defining and using Functions</li> <li>• Exception handling</li> <li>• Regular Expressions</li> </ul>	Built-in functions, customized functions Inbuilt exceptions
Day 10-12	<ul style="list-style-type: none"> <li>• OOP</li> </ul>	Class, Methods, instance creation
Day 13-15	<ul style="list-style-type: none"> <li>• Data science related packages</li> </ul>	NumPy, Pandas, SciPy, Matplotlib

As you can see, many conceptually larger topics have been crammed into very short time buckets. That's the whole point. We as experienced programmers don't need to relearn concepts such as if-else or for/while loops again from scratch, do we? All we need to learn is how things are done in Python. Idea was not to start from ground zero but leverage programming experience and learn all essential things as quickly as possible.

As I look back on my journey of learning Python in 15 days, I can really think of only 3 Key enablers that made it possible. Of course, Programming experience being implied.

**1. Goal based learning:**

- As allured to earlier, having clearly defined goal helped me create a concise learning plan that would cover all essential elements of Python and serve as foundation for Data Science specific exploration later.
- This approach will help you stay focused and not wander off, as there are just so many things to do in Python

**2. Efficient use of time:**

- Being a working professional finding enough time was always a challenge. I allocated 2 hours on weekdays and 3-4 hours on weekends, Mostly I was able to stick to schedule but if I missed, I compensated for it over the weekends.
- It was more of coding and less of reading. I remember on so many occasions I just lost track of time when I was trying something new or coding in Python. Sheer ecstasy of learning something new was at its zenith.

**3. Practice is the key:**

- More the merrier is so apt here. However, the catch is meaningful. Don't just try out the examples in the book or internet, and don't go googling the first instance you get an error, try and resolve it on your own
- I wanted to be able to do all the things in Python that I could using SAS. So, I did lot of coding practice by emulating my routine SAS programs in Python. This helped me understand similarities and weigh the differences between the two tools
- The online practice platforms I used were of immense help, they made me think and use my learnings in so many ways. "Keep coding and keep making mistakes" was my motto

## **BUTTERFLY: “INTO THE OPEN”**

By now I had completed my 15 days plan, having gone through the preparation, the feeding or learning and the transition. This approach helped me build strong foundation very quickly, it acted as a base for my future learning. I was happy with what I had achieved, and I wanted to continue this journey into the open world Python programming. Although I learnt most of the things needed for Python programming in 15 days, I continue to learn and explore the language even today. I make mistakes, which only means I continue to expand my knowledge horizons.

I am also going into the deeper aspects of SciPy, Matplotlib which I couldn't during those 15 days. These days I am learning Django and hope to create my own application soon. I also teach Python: Fundamentals course to beginners in my organization as I believe 'The extent of our knowledge grows as much as we share it'.

## **CONCLUSION**

- Python is simple yet very powerful tool, it can do all the things that SAS can and much more
- 15 Days are NOT enough to learn Python *completely*
- 15 Days are certainly enough to learn it with a specific goal
- It's easier if you have strong programming background
- For working professionals, it's difficult but not impossible
- You can learn **ANYTHING**, if you have enough zeal and right approach

## **ACKNOWLEDGMENTS**

I would like to express my deep sense of gratitude towards Shobhit Shrotriya and Supriya Moghe from Accenture for all the support and words of encouragement.

## **CONTACT INFORMATION**

Your comments and questions are valued and encouraged. Contact the author at:

Vikas Digambar Dhongde

Accenture Solutions Pvt. Ltd.

LBS Marg, Vikhroli(W),

Mumbai – 400063, India

Work Phone: +91 9819799414

Email: [vikasdhongde@gmail.com](mailto:vikasdhongde@gmail.com)/ [vikas.d.dhongde@accenture.com](mailto:vikas.d.dhongde@accenture.com)

Brand and product names are trademarks of their respective companies.