These are roughly in the order I’d try to learn them. You’ll need to set some stuff up to start programming and learn some basics about using the Python language. But once you feel comfortable with the initial stages, you should be able to hit the ground running.

Just think about what sorts of things you want to learn – personal projects and / or data you care about and/or have an interest in tend to be better motivators than generic “hello world” stuff. Also, if you ultimately do decide to put together a technical blog, personal projects are a lot more interesting for people to read about.

1. Setting up your environment

You’ll want to get familiar with setting up the tools you’ll need to code.

* 1. Pick an IDE (basically the interface you’ll do your programming in)
     1. I use PyCharm and VSCode
     2. [First Project in PyCharm tutorial](https://www.jetbrains.com/help/pycharm/creating-and-running-your-first-python-project.html)
  2. Familiarize yourself with the Linux Terminal
     1. You’ll eventually need to know how to programmatically look at your files and install software.
  3. Learn how to use a Jupyter Notebook
     1. Notebooks are convenient ways of writing code and looking at results quickly in a way that is “interactive”
     2. <https://jupyter.org/install>

1. Learn some basic Python (This is where Coursera Data Course Comes in)

A Coursera (or any other “intro to python”) course here would be helpful.

* 1. Learn about the Python syntax
  2. How to install and import Python libraries
  3. How to use library methods
  4. How to “run” some code you’ve written
  5. What the debugger is and how to use it
  6. <https://docs.python-guide.org/>, <https://python101.pythonlibrary.org/>,

1. Learn some data structures and algorithms

In the long run, it’s unlikely you’ll interface with these directly that often (unless you go into a technical engineering role), but having a sense of what data structures are, and how they’re being used under the hood of almost all code is incredibly useful and will give you much more confidence in your programming. Most object-oriented programming builds higher levels of abstraction on top of DSAs, so have a good sense of how that deep layer works is (in my opinion) important. DSAs are programming-language agnostic – e.g. a stack in Python is the same as a stack in Java is the same as a stack in C++ etc. -- they may look different because they’re coded in different languages, but they mean the same thing, and their utility is the same.

* 1. <https://www.geeksforgeeks.org/python-data-structures-and-algorithms/>
  2. <https://nibmehub.com/opac-service/pdf/read/Data%20Structures%20and%20Algorithms%20in%20Python.pdf>
  3. <https://kentdlee.github.io/CS2Plus/build/html/index.html>
  4. Some basic DSAs: lists, stacks, queues, search trees, hash maps

1. Some intro topics you might want to try out
   1. Object-oriented programming -- what are classes and how do you use them?
   2. Write your own functions or make your own mini library, and use it later
   3. Load a CSV file, look at the data, and plot it
   4. Write some functions to transform data or apply calculations to it
   5. Some tools in the Python data stack: Numpy, Pandas, Matplotlib, Scitkit-Learn
   6. <https://wesmckinney.com/book/>,
2. Other topics down the line
   1. SQL
      1. How do you set up a database, add data to it, access data?
      2. SQL is a whole different language, but it let’s you interact with data very effectively
      3. Some tools: SQLite, PostgreSQL, MySQL
      4. https://a-gentle-introduction-to-sql.readthedocs.io/en/latest/
   2. APIs
      1. What are they, why would you use them, how do you build them?
      2. Some tools: Flask, FastAPI, Django
      3. https://www.geeksforgeeks.org/python-api-tutorial-getting-started-with-apis/
   3. Basic cloud stuff
      1. Eventually you’ll need to be familiar with cloud stuff. AWS has a free digital and online skills center where you can learn the basics of their cloud solutions. I’m sure Microsoft (Azure) and Google (GCP) have similar offerings
      2. Free AWS “Becoming a Cloud Practitioner” series (4 parts)
         1. <https://www.aws.training/SessionSearch?pageNumber=1&courseId=150492&deliveryMethod=2&countryName=US&cityName=Seattle>
         2. <https://www.aws.training/SessionSearch?pageNumber=1&courseId=152373&countryName=US&cityName=Seattle&deliveryMethod=2>
         3. <https://www.aws.training/SessionSearch?pageNumber=1&courseId=152702&deliveryMethod=2&countryName=US&cityName=Seattle>
         4. <https://www.aws.training/SessionSearch?pageNumber=1&courseId=152586&deliveryMethod=2&countryName=US&cityName=Seattle>

General takeaway

I’d get very comfortable Googling “how to do X”. Also, tools like ChatGPT can be really useful for some types of questions, but be wary of the solutions you get. They’re often vague, and it’ll sometimes “make up” code that doesn’t exist and likely wont work.