

American Concrete Institute (ACI)-UIUC
Student Chapter

Python Workshop (Basic and Advanced)

Sushobhan Sen

Doctoral Candidate

April 6, 2019

Urbana, IL

Copyright © 2019, Sushobhan Sen

This work is licensed under a Creative Commons “Attribution-NonCommercial-ShareAlike 3.0 Unported” license.



Contents

1	Introduction	3
1.1	Why Python?	3
1.2	Learning Objectives	3
1.3	Installing Python	4
2	Variables in Python	6

1 Introduction

1.1 Why Python?

The Python programming language was conceived in the 1980s and the first implementation was deployed in 1991 by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands. Python is designed to be a high-level, interpreted, general-purpose computer programming language with exception-handling and the ability to be extended by users. Crucially, Python is an open-source language, meaning anybody can see the code behind it. Python today is a very popular language is a broad variety of disciplines: machine learning, web development, Geographical Information Systems (GIS), engineering, database management, high performance computing, etc.

Python 2.0, which truly launched the popularity of the language, was released in 2000, with the last version in the series, Python 2.7, set to be phased out in 2020. The current recommended standard as of when I wrote this is Python 3.7.2, which is crucially **not backwards compatible** with Python 2.x. All subversions of Python 3.x are backwards compatible though, so feel free to update regularly.

(The above information was taken from Wikipedia)

The beauty of Python is its ease - the syntax is extremely simple as compared to many other popular programming languages. It is often said, not entirely without truth, that the difference between Python code and pseudo-code is merely the indentation. Thus, the language is very easy to learn for programming novices. However, its ease should not be misconstrued to mean that it is only for easy or trivial applications: Python is an extremely powerful language with a wide variety of applications, it has extensive documentation, and its open-source nature ensures that new features are being developed continuously while bugs are also being fixed.

1.2 Learning Objectives

This workshop is broken up into two sessions:

1. A Basic Python session for beginners with zero knowledge of programming in any language

2. An Advanced Python session for those with prior knowledge of Python programming

At the end of the **Basic Python** session, participants will be able to:

1. List the types of Python variables and define them
2. Add control statements (**if-then-else**) and **for** loops to their program
3. Define and use functions
4. Define and use object oriented programming

At the end of the **Advanced Python** session, participants will be able to:

1. Use the **numpy** library to define and use matrices, and read and process data from files
2. Use the **pandas** library to read and process data
3. Use the **scipy** library to solve a system of linear equations and implement other useful scientific functions
4. Use the **matplotlib** library to create publication-quality plots

The lists above were what I will try to accomplish in our two two-hour long sessions. Depending on how quickly the workshop goes, I may or may not be able to meet all the learning objectives. However, I will get you along far enough so that you can complete any remaining items on your own.

1.3 Installing Python

The easiest and **recommended** way to get all popular Python libraries and IDEs is by downloading and installing the latest Anaconda distribution for your computer. Make sure to download the latest version corresponding to Python 3.x.

Alternatively, you can install and use Python using the terminal:

- On Windows 10, activate Windows Subsystem for Linux

- On Mac OS or any UNIX-like OS (such as Linux), just run your favorite Terminal application

Then install Jupyter with `pip`. With any other version of Windows (which you should not be using for too long on your personal computers anyway for security reasons), Anaconda is your best option.

Once installed, you can now create a new notebook. If you installed Anaconda, open the Anaconda Prompt, navigate to your directory, and use the `jupyter notebook` command. If you choose to use the terminal, follow the same steps on the terminal. Both methods will launch Jupyter Notebooks in your browser. From there, you can create a new Notebook.

If you prefer not to install anything on your computer but would rather run Python remotely from your browser, you can use the Online IDE from repl.it. This doesn't always work very well though. Any other online IDE that you find should be OK too.

2 Variables in Python

Python defines the following types of variables:

Variable	Syntax	Description	Comments
----------	--------	-------------	----------

```
for i in range(0,1):  
    print(i)
```