SciPy 2019
July 8 - 14
Austin, Texas

Setup Your Workstation

Due: February 15, 2019

SHORT BIO

Abeer Eltanawy: Born and raised in Egypt, started out as a chemist, trained as a research scientist in Cancer research where I finished my master's in Cancer Research. In 2017, I transitioned in a software company called Schrödinger, a scientific leader in developing chemical simulation software for use in pharmaceutical biotechnology and materials research in the cheminformatics field. Currently, I'm an Education Specialist working from the company's Mannheim office in Germany.

Monica Chelliah : Monica currently is a software developer at Schrodinger, where she builds front-end python applications. An avid baker, she can frequently be found testing new recipes on unsuspecting coworkers.

PREREQUISITE SKILLS

Absolute beginner in Python and development practices. Ideal for teen track.

TUTORIAL DESCRIPTION / ABSTRACT

- Target audience
    - Beginners / teen track
- Expected level of knowledge prior to the class:
    - Familiar with command line (UNIX/Bash)
    - Familiar with Jupyter Notebook by taking a quick look at the attached documents.
    - Go through the IDE chapter and install their favorite one.
    - Be prepared with the attached material and any prerequisite steps.
- Goals of the class
    - Gain an understanding of a typical development workflow using virtual environments, version control, and best practices.
    - How to write effective object-oriented python, and practice test driven development.

DETAILED OUTLINE

Jupyter Notebooks (Included in tutorial repo. User to read before the tutorial day)

- Jupyter Notebook Summary
- Jupyter Notebook cheatsheet

Introduction (10 min)

- Why taking the time to setup your workstation is a good idea?
- A basic outline of what the tutorial is going to cover.
- Any caveats for different operating systems - Mac OS / Linux / Windows

Python (10 min)

- Which Python interpreter should I use?
- Python 2 or 3? What does the official guideline recommend?
- A quick run down through the existing / common interpreters
- Link to official python installation to download (should they have already done this / would it take too long to download and install at this point?)
- PEP 8 - Style Guide for Python Code

Virtual Environments (5 min)

This section will be conducted using the virtual_env jupyter notebook that can be found in this tutorial repo.

- What and why are virtual environments useful?
- Create a virtual env - activate and deactivate environment with an example package

Pip (5 min)

The audience will view requirements.txt file, and will be instructed to download required libraries and dependencies using the following command in their shell:

```
pip install -r requirements.txt
```

- What's pip?
- Install some common packages we'll be using later in the tutorial

Version Control Using Git (45min)

This section will be hands-on practice using a typical git development workflow using the terminal.

- What is Git? How does it work?
- Go through common terminology: commits, log, branches, repo, etc.
- Do some simple exercises with commits and branching
- Hands-on - an example of a typical git development workflow:
    - Ask participants to create a test repo using their GitHub
    - Create a local branch
    - Edit a local file and commit the changes
    - Merge the changes into master
    - Push the changes upstream

Break (15 min)

- Catch up opportunity for anyone who is lost or questions

Text Editor vs IDE (Distributed)

- Difference between text editors and IDEs
- PyCharm, Atom, vim
- Go through some common ones for both
- Ask to download their favorite before the tutorial day

Object Oriented Programming (1hr 45min)

This section will be hands-on training using shell and the user favorite editor. The user will experience writing simple methods and functions.

- Classes
    - Explain the objective of using OOP
    - Go over what a class, instance and method objects are
    - Hands-on:
        - Write a python class
        - Understand the difference between a method and function
        - Test the difference between a class and instance variables
- Inheritance
    - Hands-on:
        - Write a subclass inheriting from the previously written class
        - Understand the requirements of how to access methods from superclass
- Context manager

    - Why are they useful?
    - Hands-on:
        - Reading and writing to a file
- Decorators
- Useful modules: json, csv, pathlib, subprocess, os
- Generators & yield?

Testing Your Code - Test Driven Development (45min)

This section will be hands-on training using terminal and the user favorite editor. The user will experience writing tests using TDD principles. Also will write a test for a method/function conducted from the previous OOP section.

- What is Test Driven Development?
- Importance of testing your code - example
- Pytest:
    - How to test (writing a simple test)
    - Testing expected failure
    - Parametrization
    - Fixtures

INSTALLATION INSTRUCTIONS

As this tutorial aims to teach and help participants to setup their working environment, the only requirement is to have a working installation of python and a GitHub account. We will provide a short script for participants to verify that their machine has python installed.

check_version.py:

```
import sys

if (sys.version_info > (3, 0)):
    # Python 3
    print("Awesome, you're all set to go!")
else:
    # Python 2
    print("Please install Python 3.")
```

LINKS TO IN-PROGRESS TUTORIAL MATERIALS

Tutorial github repo