PyStarters

Charly, Steve, George, Francesca

2018

E-mail: pyclub@ucl.ac.uk & k.otis@ucl.ac.uk Github repo: **https://github.com/swc-pyclub/pystarters**
Class Hours: 9:30-11:00
Class Day: Tuesday
Class Room: See detailed schedule

Course Description

Efficient programming in Python using good coding practices.

Required Materials

- The course assumes basic knowledge of Python that is easy to find elsewhere
- Laptop with wireless connection to eduroam

Course Objectives

Successful students:

1. Setup a python development environment everywhere.
2. Write high quality python.
3. Solve problems relating to data analysis and data manipulation.
4. Develop the required knowledge to join the Pyclub.
5. Develop a common Python programming style to the center with the end goal of contributing to a shared software library.
6. Have fun.

Course Structure

Class Structure

The sessions will interchange between a tutored session (odd weeks) and a un-tutored one (even weeks). The assignments are given without precise instructions at the end of the previous tutored session and the pystarters will have one month including the next unsupervised meeting to come up with a consensus solution. This solution will then be discussed at the tutored meeting after introducing the best approach.

Assessments

Series of assignments to complete for the next tutored meeting. The attendees should use the even week number meetings (untutored) to work towards the solutions in groups and present their consensus for the next tutored meeting.

Final Exam and Class Project

A significant functional codebase but breaking all rules of good coding practices will be handed at the beginning and attendees will have the whole year to refactor it to a clean structure. Attendees who complete the assignment will become TAs. A complex project will also be undertaken collectively throughout the year.

Schedule and weekly learning goals

Week 01, 27/02: Introduction to the pyclub, Python and the programming tools used in the class

- Learn about the pyclub and the program for the year.
- Understand the objectives of PyStarters.
- Basic style - writing *pythonic* code (python3, import/packages, modules, enumerate . . .) to ease sharing.
- Basic tools (pycharm, jupyter . . .).
- The python interpreter.
- The python documentation.
- Setting up a python environment (hands on).
- Introduction to next week's exercises

Weeks 02 - 03, 13/03 - 27/03: Built in Python data types and Jupyter.

- Learn about environments.
- Work with anaconda.
- Master Jupyter.
- Learn to manipulate python built in variables efficiently.
- Learn about ducktyping in python (e.g. for iterables)
- introduction to the STL ?

Weeks 04 - 05, 10/04 - 14/04: Version control

- Introduction to the concept of VCS.
- Install and learn basic git usage.

Weeks 06 - 07, 24/04 - 08/05: Functions and Procedural programming

- Modularising programs into functions
- Pass by value vs pass by reference
- The python import system
- Docstrings

Weeks 08 - 09, 29/05 - 12/06: Python science libraries

- Numpy (efficient array operations)
- Scipy

- Matplotlib (graphics)

Weeks 10 - 11, 26/06 - 10/07: Follow up of week 5
Weeks 12 - 13, 24/07 - 14/08: Object oriented programming

- Introduction to OO programming in Python
- Special methods and operator overloading

Weeks 14 - 15, 28/08 - 11/09: Exceptions and testing

- Exception handling
- Defining custom exceptions
- Introduction to software testing
- Unit testing with pytest

Weeks 16 - 17, 25/09 - 16/10: Packaging

- distutils
- setuptools
- conda recipes

Weeks 18 - 21, 30/10, 13/11, 27/11, 11/12: Selection of topics between GUI programming, profiling and optimisation, scripting, R in python for statistics