MSc in Bioinformatics for Health Sciences
PYT. Introduction to Python

Syllabus Information

Academic Course: 2018/19

Academic Center: 804 - Official Postgraduate Programme in Biomedicine

Study: 8045 – Bioinformatics for Health Sciences - MSc

Subject: 31030 – PYT. Introduction to Python

Credits: 5.0

Course: 1st

Teaching languages: English

Teachers: Javier Garcia

Teaching Period: 2nd term

Presentation

The main goal of the course is to learn the basic concepts on programming with Python, and to learn how Python features can be used to deal with biological research data. The subject is concerned with providing students with the tools to create, understand and maintain complex software in a productive manner. Starting from basic programming procedural constructs, the students will be introduced to object oriented techniques to produce code which is clear, understandable and well documented. This subject is highly recommended for students wishing to be able to develop their own software in a collaborative environment. A collaborative project will be developed, in which the students will have the chance to implement a complex software for solving specific problems presented in other subjects. This subject is completely incremental. Consequently, it is strongly advised that students keep up to date with class material.

Associated skills

The main objective is the student achieves the following abilities:

Instrumental

1. Expertise in:
    1. Procedural Python
    2. Reading and interpreting Python documentation
    3. Writting and documenting Python code

2. Understanding of Object Oriented Programming:

    1. Ability to factor a problem into objects.
    2. Understand how Python implements the object oriented concepts.
    3. Development of software using Python.

<u>Interpersonal</u>

1. Ability to solve a biological problem with computational programming techniques using Python.

2. Ability to develop a collaborative software project within a working group

<u>Systemic</u>

1. Leading capacity on a working team and motivation to finish on time the work.

2. Creativity and motivation to increase the quality of the produced software.

Prerequisites

It is highly recommended that students have at least the competencies provided by the “Introduction to Algorithmics” and “Introduction to Perl”. It is also recommended a solid understanding of UNIX file system. Previous knowledge of Object Oriented Programming or knowledge of Python is not required.

Contents

Separation into blocks and their duration is just for reference. There is not a clear separation of themes nor their weight in the final grade.

Block 1. Introduction to Python.

– General introduction to programming.

– Built-in types and functions.

– Control structures.

– Definition of functions.

– Introduction to modules.

– Analysis of algorithms.

Block 2. Object Oriented Programming.

– Classes and Objects.

– Inheritance.

– Relations between classes.

– Naming conventions.

– Introduction to UML.

– Introduction to software engineering.

– Management of exceptions.

Block 3. Python Libraries.

– Use of standard Python modules.

– Creation of a library.

– Creation of documentation.

Block 4. Third-party libraries.

– BioPython.

– Numpy

– Scipy

– Matplolib

– Graphical interfaces: TKinter.

Final project

The final project will consist on programming a standalone program for solving specific problems presented in other subjects. The possible projects will be presented at the beginning of the course. Students are encouraged to work on the project from the beginning of the course, progressively introducing the new concepts acquired in the classes and with the exercices.

Teaching methods

Theoretical concepts will be introduced during presential lectures. Online material will be available online. After the theoretical lecture of each session, a related activity will be proposed to the students. In each session students will have time to work on the problem and to ask doubts on the introduced concepts. The problem solution must be submitted individually during the same session or just the day before the next session, depending on the activity (continuous evaluation).

Programme of activities

Estimated time spent on the subject:

- In the classroom: Work in the classroom includes theoretical lectures followed by personal time in which students must solve proposed activities. (30 hours in total)

- Outside the classroom: Activities outside the classroom include the finalization of proposed activities and the development of the final project into groups of 2 people. (95 hours)

Evaluation

Assessment system:

The evaluation will consist on two blocks:

1) Continuous evaluation (50%). Individual.

50% At the end of each session, the student must submit individually different Python scripts that will be automatically evaluated. Evaluation will take into account:

1. The script runs under different input conditions without producing errors.

2. The script output is equal to the expected output.

3. Performance of the script.

50% Face-to-face exercise. During the last session, different exercises will be proposed. Exercise must be submitted at the end of the session

2) Final project (50%). The class will be divided in different groups composed by students, and each one will have a project to develop. All the groups will have the same topic. The projects will be related to some of the areas of expertise of the master. Code generated in continuous evaluation submissions can be used in the final project, as well as the use of standard Python libraries are encouraged to be used. Projects will be proposed at the beginning of the course, and it will be shared with the subject Structural Bioinformatics. Evaluation will take into account: program structure (tasks, classes, modules,...), reusability and usage of libraries, graphical output when possible, the documentation and the benchmark used.

Bibliography and Information Resources

Think Python. How to Think like a computer scientist. Allen Downey. Green Tea Press. Needham, Massachussets (http://www.greenteapress.com/thinkpython2/index.html). Available for download at:
http://www.greenteapress.com/thinkpython2/thinkpython2.pdf

Python Cookbook, 3rd edition. Recipes for Mastering Python 3. By David Beazley, Brian K. Jones. Publisher: O'Reilly Media. May 2013