

# Introduction to Computation and Programming with Python

2021/2022

## Prerequisites to Enroll

None

## Prerequisite reading

None

## Overview and Objectives

The aim of this course is to provide an introduction to programming and the Python language. Students are introduced to computation and complexity, learn the core programming concepts, and start managing the most common libraries in the scientific and data analysis fields. The main idea of this course is to get students coding Python quickly. Since companies worldwide are using Python to harvest insights from their data and gain a competitive edge, this course also provides very initial concepts on Machine Learning through Python.

The pedagogical philosophy in this course embraces the principle of learning by doing. Lectures will give an overview of each topic and students will learn through practical hands-on coding exercises. After finishing each of the topics, a questionnaire is used to reinforce the contents. With all this knowledge, students should develop a final Python project to demonstrate their mastery of all the concepts learned.

## Course Outline

1. Computational thinking (2h)
  - Abstraction, decomposition, algorithm, pattern recognition, representation
  - Computation
  - Flow diagrams, pseudocode, and high-level code
2. Programming languages (1h)
  - Interpreted vs Compiled
  - Why Python
3. First steps with Python (6h)
  - Jupyter Notebook
  - Variables, data types and casting
  - Conditionals
  - Loops
  - Functions
  - File handling

- Advanced data structures: Tuples, Sets, Dictionaries
4. Algorithms (3h)
    - Complexity:  $O$ ,  $\Omega$ ,  $\Theta$
    - Searching mechanisms
    - Sorting mechanisms
  5. Managing common libraries (4h)
    - NumPy: Working with arrays
    - Pandas: Analyzing data
    - SciPy: Scientific Python
    - Matplotlib: Graph Plotting
  6. Introductory concepts on Machine Learning with Python (4h)
    - Mean, median and mode
    - Standard deviation
    - Data distribution
    - Linear and Non-Linear Regression
    - Decision Trees
    - Scale, train, and test

### **Required Activities**

Exercises, questionnaires, and a final project.

Class participation is compulsory. During classes there will be little exercises that students are expected to perform while in class. There will also be larger exercises that students are expected to solve in their own time.

After finishing each topic, questionnaire is used to reinforce the contents.

A final Python project is required to show the knowledge acquired.

### **Evaluation**

Homework exercises (50%), Follow-up questionnaires (10%) and Final Project (40%)

### **Competences**

(forthcoming)

### **Learning Outcomes**

- Identify core aspects of programming and features of the Python language
- Model problems from a computational point of view through abstraction
- Understand and apply core programming concepts like data structures, conditionals, loops, variables, and functions
- Design and write fully-functional Python programs using commonly used data structures, custom functions, and reading and writing to files
- Understand the complexity of an algorithm and code efficient algorithms

- Manage some of the most common libraries in the scientific and data analysis fields
- Apply very basic data science techniques using Python

### **Materials**

In general, a student attending all classes will not need any book to pass this course. Online resources, practical exercises and project implementations will probably prove more useful than a book.