DEPARTMENT OF MEDICINE, HUDDINGE

H7F6034, Introduction to Programming Using Python, 3 credits (hec)

Introduktion till programmering med Python, 3 högskolepoäng

Third-cycle level / Forskarnivå

Approval

This syllabus was approved by the The Committee for Doctoral Education on 2024-09-13, and is valid from spring semester 2025.

Responsible department

Department of Medicine, Huddinge, Faculty of Medicine

Contributing department/s

Department of Medicine, Huddinge

Prerequisite courses, or equivalent

No prerequisite courses, or equivalent, demanded for this course.

Purpose & Intended learning outcomes

Purpose

Python is a programming language that:

1. is modern, easy, highly-supported, and widely used,
2. provides multiple tools to manage data, such as organized datastructures, CSV files, tools for creating advanced data visualizations, and more, and
3. is the de-facto language for artificial intelligence and machine learning applications.

The course will enable the participants to acquire a solid understanding of how programming languages work as well as concrete hands-on knowledge of Python. Their acquired knowledge will enable them to adopt recent technologies, for example technologies regarding data management and analysis, as well as artificial intelligence and machine learning.

Intended learning outcomes

After successfully completing this course, a student will be able to:

- set up and use an integrated development environment (IDE) for developing Python code
- develop Python programs that solve tasks (relevant for students and researchers)
- automate (through creating Python code) data manipulations, dataset cleaning and curation, aggregating data and extracting statistics, create visualizations such as graphs and plots
- handle advanced data structures such as arrays, matrices, and trees
- use the IDE to swiftly find problems and mistakes in their code and correct them
- efficiently search and find answers and solutions on how to implement specific parts of code

Course content

The course starts by setting up a working environment for Python; this enables students to directly test and experiment with everything they learn during the theoretical lectures. During the first half of the course, fundamentals of programming are introduced, such as constants, variables, memory operations, arithmetic, and logical operations, followed by the principles of structured programming [1] (i.e., branching logic and loops). These concepts are explored with examples and exercises/projects that enable participants to familiarize themselves with the real-life application of Python in science. In the second half of the course, the more advanced concepts of functional and object-oriented programming [2] are introduced, again with practical Python examples.

Forms of teaching and learning

Morning hours include lectures that introduce new concepts and material (9 lectures). Afternoon hours include practical exercises where students are given a task every 2 or 3 days to solve, with help from the teaching personnel. Teaching personnel will be available during evenings to provide support, answer questions, and guide students in correctly tackling each task. In total, 3 such tasks are given over the duration of the course. On the last day, students are given a final project task to work on individually in the morning (without help from the teachers) and are expected to present their solution in the afternoon of the same day. This task is the basis upon which students are graded (see also Forms of assessment).

Language of instruction

The course is given in English

Grading scale

Pass (G) /Fail (U)

Compulsory components & forms of assessment

Compulsory components

All the following requirements must be met:

- participation in at least 7 out the 9 morning lectures
- submitting a solution to each of the 3 tasks
- participation during the final day
- submitting and presenting a solution for the final project task

Absence from lectures can be compensated by finishing an additional task.

Forms of assessment

Project (i.e., the task of last day) presentation and review. The student will be asked to present how they solved the task, how they designed their solution, asked to explain design choices, and asked to explain the basic principles of some of the components they chose to use.

Course literature

Literature recommendations (not required):

Theoretical

[1] Ole-Johan Dahl etc., “Structured Programming”, Academic Press Inc, 1997, ISBN-13 978-0122005503

[2] David Wood, “Head First Objects-Oriented Analysis and Design: The Best Introduction to Object Orientated Programming”, O’Reilly Media, 2006, ISBN-13 978-0596008673

Python specific

[3] Mark Lutz, David Ascher, “Learning Python 2e”, O’Reilly, 2004, ISBN-13 978-0596002817

[4] Stephen Gruppetta, “The Python Coding Book: A relaxed and friendly programming textbook for beginners”, 2024, ISBN-13 979-8884788176

[5] Abhishek Singh, “Master Python Using Version 3.11: Learn Python Like Never Before”, 2023, ISBN-13 979-8385523276

Recommended resources for python

Python language:

- miniconda, https://docs.anaconda.com/miniconda/

Integrated Development Environment (IDE):

- PyCharm, https://www.jetbrains.com/pycharm/