MACIEJ WIELGOSZ, MARCIN PIETROŃ

PROGRAMMING IN PYTHON

September 2020

Exercises

Input/output

Hello World	Write a script that displays the greeting (e.g. "Hello World") with and without *shebang* (running it with the standard interpreter command - python3)
Inputting data	Write a script that asks for your first name, last name and year of birth (should be on one line)
Data saving	Write a script that implements the code lock function. It asks for the code and then checks if it matches the previously entered code

Notes

Working with files

Files count	Write a script that counts the number of files in the */dev* directory, use the standard library - os
Directory's structure	Write a recursive transition of the directory tree and list the files that are in the structure being explored
Extensions converting	Write the script to convert the file extensions from **.jpg* to **.png* (first create a set of 4 files with the extension **.jpg*)

Notes

TEXT

Words removal

Write a delete script from the input text string (select some files from the Text repository) the following words: *się, i, oraz, nigdy, dlaczego*

Words replacement

Write the change script in the provided input string (select some files from the Text repository) the following words: *i, oraz, nigdy, dlaczego* with the following set: *oraz, i, prawie nigdy, czemu*. The recommended structure is a dictionary.

NOTES

Calculations and algorithms

Quadratic equation	Write a script that calculates the roots of the quadratic equation in the form : $y = ax^2 + bx + c$. The script input values are: a, b, c
Sorting	Write a script to sort numbers in descending order. Randomly generate 50 numbers - use the standard randomization function. Use the built-in sorting function only to verify the results
The scalar product	Write a script that calculates the value of the product of two vectors: $a = [1, 2, 12, 4], b = [2, 4, 2, 8]$, the so-called scalar product of vectors
The sum of the matrix	Write a script summing two matrices of 128x128 size. Use a random number generator to generate a matrix
Matrix multiplication	Write a script that performs the multiplication of two 8x8 matrices
Determinant of the matrix	Write a script that calculates the determinant of a randomly generated matrix

Notes

Classes

Complex numbers	Define a class representing complex numbers (along with functions acting on them, e.g. addition, subtraction, etc.)
Calculator	Use the class above to create a simple calculator that parses and executes the equation given by the user

Notes

Working with data

XML	Parse the prepared XML (with SAX and DOM parser) and modify it, e.g. change the value of a tag and save it to a new file
CSV/JSON	Write a program that asks the user for data containing several fields (it can be, for example, a list of tasks with a description and due dates, or a database of movie reviews) and saves the given data to a file in a selected format (CSV/JSON). Each time the program is started, the program should read and display the previously entered data, make it possible to delete them (one entry each) and add new records.

Notes

Parallelization of calculations

Histogram	Implement a multi-threaded histogram count (monitor the execution in htop)
Five philosophers	Implement a simple five philosopher problem (with deadlock), then remove deadlock.

Notes

Final projects

Principles of work

Please use

- git repository (shared with the lecturer),
- PyScaffold,
- VirtualEnv

Final project

- results (custom implementation) should be compared with the implementation available in *scikit-learn*,
- the finished student project should contain full documentation on the implemented algorithm and tests,
- each student team may receive a code from another team to test and evaluate.

Notes

Available project topics

	• DB scan, • Birch, • SVD, • Random Projection, • SVM, • Learning Vector Quantization (limited availablity in *scikit-learn*), • Random Forest, • Cart, • Hierarchical Clustering, • k-NN, • Affinity propagation, • Aglomeratitve Clustering, • Gaussian mixtures, • K-Means, • Mean-shift, • Nearest Neighbors, • Naive Bayes, • Decision Trees. Sample implementations of algorithms are available in the library scikit-learn.
Learning and test data	• Text • Images

Notes

Stages of work

1. forming a 3-person team (suggested: 2 programmers, 1 tester and repo administrator),
2. selecting a project topic,
3. establishing a repo and establishing a project development strategy within its framework,
4. analysis of the selected algorithm,
5. implementation,
6. testing - comparing the results of the implemented module with the version available in *scikit-learn*,
7. development of project documentation (both algorithm and code)

Partially the last two steps (testing and code documentation) are available in *PyScaffold*: proper code documentation (comments in the correct format) and writing tests are required.

Notes