

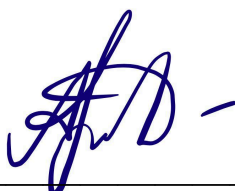
NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Applied Mathematics and Informatics"

Software Project Report on the Topic:
Development of Materials for Teaching Programming in Python

Fulfilled by:

Student of the Group
БПА/Д212
Dekkusheva Alina
Vladimirovna



(signature)

May 31st, 2024
(date)

Checked by the Project Supervisor:

Voznesenskaya Tamara Vasilievna
Associate Professor
Big Data and Information Retrieval School, Faculty of CS, HSE University

(signature)

June 2nd, 2024
(date)

Moscow 2024

Table of contents

Table of contents.....	2
Abstract.....	3
Introduction.....	4
Relevance.....	5
Expected outcomes.....	6
Instruments used.....	6
Subject area.....	7
General process.....	8
Basic Terms and Definitions.....	9
Working process.....	11
Comparative analysis of sources and analogues.....	18
Results and conclusions.....	21
Main results.....	21
Prospects of further work.....	22
Conclusions.....	23
List of sources.....	24

Abstract

This project is detailed on the development of educational materials for teaching programming to students. The work was done in the Jupyter Notebook and the Yandex Contest environment, it included topics: strings, lists, NumPy.

In this work, comparative analyses were performed to determine the best educational practices for students.

Therefore the main project outcomes are: interactive educational materials, promoting student participation, in-house educational content, environment for students with activities to be performed. The existing content will be made better, getting the more advanced topics implemented, multilingual support for students who want to learn the course in a different language and ongoing improvements based on the feedback and python updates.

This project provides an essential foundation for students eager to learn Python and achieve programming skills.

Introduction

As software development continues to change, programming is becoming increasingly essential. Python is an excellent all-purpose coding language because it is both so easy to pick up and has a huge number of shortcuts and libraries to make virtually any kind of application with.

In addition, we use this project to introduce students to programming and provide good foundations for their oncoming programming education. This is implemented through Jupyter Notebook and Yandex Contests.

The project provides a comprehensive set of materials for learners which will guide them in the learning process.

Relevance

Reasons to provide educational materials for teaching Python programming has grown due to following reasons in the recent times:

- **Demand for Coding Abilities Growing:** In many industries, coding is becoming the new literacy. Python has been adopted by developers and professionals alike, since it is versatile and has powerful implications at the same time.
- **Potential performative activity:** Our project as a further outreach to live learning is to bring live learning to students where they can take part in the material using Jupyter Notebooks. This way students learn complicated ideas.
- **Career Ladder To Further Work:** This way a student can climb the ladder from a beginner level to an expert level.

Expected outcomes

This project is expected to yield Educational material on Python Programming in the areas on:

1. Lists
2. Strings
3. Numpy

Instruments used

During the production, the following tools were used:

1. Jupyter notebook
2. Yandex Contest
3. Google Drive
4. Google Slides
5. Google Docs
6. GitHub

Subject area

This project is complementary to the creation of tutorials, guides and Jupyter notebooks for teaching Python programming, designed for beginners and intermediate learners. It is a convergence of a host of fields, the main ones being:

1. **Getting Started with Pythons:** The project is for beginners. It teaches some basic concepts of python and this is how to get started. It covers the following areas which will help students to gain knowledge about Python and a bit of a more complex topic in python which is NumPy.
2. **Data structures:** The project includes basic data structures like (and mostly comprises of those) Strings and Lists in Python. They are core programming concepts that facilitate storage and manipulation of the data in an efficient manner. Knowing these data structures is important while writing respective code more efficiently and practical.
3. **NumPy:** Library for numerical computations in Python. It supports a wide range of large, multi-dimensional arrays or matrices and extensive mathematical functions to operate on these arrays. This is especially useful for students who are considering working on data science courses as NumPy is a staple for data analysis and computational work.
4. **Interactivity:** The project is made up of Jupyter Notebooks and therefore leads to a very interactive learning environment. Students are able to learn this way due to this interactive format.
5. **Instructional pedagogy:** The project aligns with current instructional practices of tutorials, coding practice, and coding exercises.

General process

How the basic stages work:

1. **Preliminary research and planning:** carried out an extensive analysis, seeking motivational resources and the best of teaching skills in terms of Python Programming Courses that are available. So established specific leanings that were to be accomplished from each subject.
2. **Content development:** wrote step by step a detailed instruction for each topic (NumPy, strings, lists). This consisted of preparing sections from writing explanations, designing examples, to creating interactive exercises. Structured content progressively.
3. **Interactive Jupyter Notebooks/Yandex Contest:** created Jupyter Notebooks for each topic, embedding the instruction and the interactive exercises. Implemented Yandex Contests for themes.

Basic Terms and Definitions

- **Python** - a high-level, general-purpose programming language famed for its readability and versatility. Procedural(object-oriented) Language and Python (a Functional Programming).
- **Data type** - It is a categorization that defines which kind of value a variable holds like: integers, strings or lists.
- **Function** - a self-contained block of code functionality that accomplishes a particular task.
- **Methods** - a function is associated with an object, it is defined inside a class.
- **Immutability** - the property of certain objects whose state can not be modified after they are created. Types like strings, tuples and integers are immutable objects.
- **Concatenation** – the method of combining two or more strings to form the single string.
- **Indexing** (also referred to as subscripting) - a way of referring to elements of a sequence (a list for example) using part of the element's position.
- **Unicode** - a standard character encoding system intended to support text representation and processing.
- **Slicing** - Changing the second element using the index of the elements.
- **Case sensitivity** - the language distinguishes between uppercase and lowercase letters.
- **Regexes (regular expressions)** - a sequence of characters used to define a search pattern - are often used to search, match or manipulate strings devices.
- **Encoding/decoding** - the process where one type of data is transformed into another.

- **Dynamic datatype** - type of a variable is decided during runtime, not before that time.
- **Iteration** - the process of repeating a set of statements, usually using loops.
- **Stack** - a data structure that obeys Last In, First Out (LIFO) principle, the last element entering the stack is the first to leave the stack.
- **Queue** - A queue is a data structure which is used for insertion and deletion of elements. It follows the First In First Out (FIFO) structure where the first element inserted is the first to be removed.
- **Attribute** - a value or a method associated with an object. Attributes attribute data and functionalities specific to a Class.
- **Aggregations** - calculations with many values to yield a single summarizing result.
- **Library** - a set of modules that implement many of the tasks needed for web programming, date and time operation, and mathematical operations.
- **NumPy** - Python library adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
- **Interactive learning** - coding practicals or projects. This cultivates active learning and aims at consolidating the knowledge of students.
- **Student guideline** - a set of instructions and rules to help students out through the struggling course of education.

Working process

As mentioned above the work on the project was divided into a few main stages with many substeps on each of them. Specific work included:

- **Review of Literature and Analysis:**
 - Review available Python courses and educational materials (section. Comparative analysis of sources and analogues.
 - Determine best practices and evidence-based principles applied to similar types of courses.
- **Define learning outcomes:**
 - Define the key learning outcomes for each topic.
 - Specify what skills and knowledge expected from the student after each module (NumPy, Strings, Lists).
- **Creating educational content for NumPy:**
 - Deliver comprehensive training resources that include:
 - array creation
 - array attributes
 - array indexing
 - array slicing
 - array mathematics
 - aggregation functions
 - random number generation
 - linear algebra operations
 - reshaping and transposing
 - Put practical questions in this Jupyter Notebook itself.
- **Create literacy content around Strings:**
 - Teach the Ultimate Guide on this topic.
 - basic properties
 - inner implementation

- slices
- in-built methods
- regular expressions
- encoding and decoding
- Add solving problems on Jupyter Notebook and Yandex Contest

A. Head over heels

Time limit	1 second
Memory limit	64.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

You have been given a word. Make its last letter the first. All the subsequent letters will be shifted to the right by one position in this case.

Input format

A line consisting of one word (a sequence of lowercase Latin letters without spaces) of no more than 200 characters.

Output format

Output the resulting string

Picture 1. “Head over heels” task.

B. А роза упала на лапу Азора...

Time limit	1 second
Memory limit	64.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

Write a program that would determine whether the entered word is a palindrome. Please note that your programme must be case-insensitive.

Input format

The input to the program is a string of no more than 20 characters, containing no characters other than numbers and letters (both uppercase and lowercase).

Output format

If the input string is a palindrome, output "YES". Otherwise, output "NO".

Picture 2. “А роза упала на лапу Азора” / “Stressed desserts” task.

C. IP?

Time limit	1 second
Memory limit	64.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

In order to access the Internet, each computer is assigned a so-called IP address. It consists of four integers ranging from 0 to 255, separated by points.

Write a programme that would determine whether a given string is a valid IP address.

Input format

The input to the programme is a string of no more than 20 characters.

Output format

If the input string is a correct IP-address, output "YES". Otherwise, output "NO".

Picture 3. "IP?" task.

D. University of Mars

Time limit	1 second
Memory limit	64.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

The National University of Mars has introduced additional standards for academic writing. Since this moment, all the students in their course works must write the numbers from 0 to 9 (1-digit non-negative) as words. (E.g. "one" instead of "1").

Write the programme that will help a student to rewrite his course paper so that it satisfies this novelty.

It is guaranteed that the paper does not contain any other numbers except for those on the interval [0; 9]

Picture 4. "University of Mars" task.

- **Create how to content for Lists:**

- Create modules for learning and using lists in python these include:
 - basic properties
 - inner implementation
 - slices
 - in-built methods
 - multi-dimensional spaces
 - list as stack and queue
 - iterations
- Add useful exercises to the Jupyter Notebook and Yandex Contest

A. Interval

Time limit	1 second
Memory limit	64.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

You are given an array A of indefinite length and two numbers - i and j . It is necessary to display all elements of the array from the i -th to the j -th (both inclusive).

Input format

The first line of the input file contains an array A , the elements of which are separated by spaces. The second line contains 2 numbers separated by a space - i and j .

Output format

Print all elements of the array from the i -th to the j -th inclusive. If this cannot be done, do not output "NO".

Picture 5. "Interval" task.

B. Grasshopper

Time limit	1 second
Memory limit	64.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

You are given an array of indefinite length and a natural number N . Output all the elements of the array whose indices are a multiple of N .

Input format

The first line contains an array whose elements are separated by spaces. The second line contains the number N not greater than 10.

Output format

Print the corresponding array elements separated by spaces

Picture 6. “Grasshopper” task.

C. Pinta

Time limit	1 second
Memory limit	64.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

Write a programme that would determine whether the sum of the squares of all the elements of an array of size N is a 5-digit number.

Input format

The first line of the input file contains a natural number N — the number of elements in the array. The next line contains N integers separated by spaces - these are array elements.

Output format

If the input data satisfies the condition of the problem, the programme should output "YES", otherwise "NO".

Picture 7. “Pinta” task.

D. Competition

Time limit	1 second
Memory limit	64.0 MB
Input	stdin or input.txt
Output	stdout or output.txt

Once upon a time, a competition was held at a school for young programmers. Now, it's time to see the results.

Each participant has his own identification number assigned and also a certain number of points he/she managed to gain. Display the tournament table - a list of competition participants sorted by the number of points scored.

Input format

The first line contains the number n - the number of participants. Each next line contains a pair of numbers separated by a space: the number of points scored and the identification number of the corresponding participant. All numbers in the input file are non-negative integers and do not exceed 1000.

Output format

Output the original list in descending order of points. If some participants have the same points, then they need to be sorted among themselves by the identification number in descending order.

Picture 8. "Competition" task.

Comparative analysis of sources and analogues

Creating the Educational materials on Teaching Python programming, the comparison of our effort with the similar resources and materials that already exists is the first most important piece. This is useful for performing comparative analysis to determine the strengths, weaknesses, and what the project does differently from the others.

1. Codim (Кодим)

Online Platform for Coding Courses for School Students — Codim has a python too.

Structure:

- **Basic Python:** variables, data types, and basic operations.
- **Advanced Python:** Functions, Loops, and Data Structures
- **Project-based Learning:** Includes practical projects such as game development and data analysis.

Comparison with our case:

- **PROS:** Codim offers an interactive platform, on a project basis and tracking progress in real time that keeps students included in the process.
- **CONS:** minor bugs in observations, however the overall experience is very positive for both the company and the user. Not as in-depth in advanced topics as our materials Our project also goes into passing detail about libraries such as NumPy, which are essential for data science applications.

2. Yandex Lyceum (Яндекс Лицей)

Yandex Lyceum offers a two-year study course specializing in Python programming for high school students.

Structure:

- **First Year:** Basics of Python: syntax, basic operations, and simple algorithms.
- **Second Year:** Advanced topics such as web development, data structures, and algorithmic thinking.

Comparison with our case:

- **Strength:** The long time span over which the program is conducted captures superior learning and sustained participation.
- **Disadvantages:** The program is long and structured for a breadth of information, so it may not be as accommodating to students interested in quicker, more intense courses. Our project's approach with Jupyter Notebooks offers more flexibility and immediate application of concepts.

3. Algorithmics (Алгоритмика)

Algorithmics is a Russian-based international kids coding school. They have different Python courses for various age groups and skills. Game Based Learning college: this attracts the student towards the school.

Structure:

- **Beginners:** Introduction to Python, basic commands, and simple projects.
- **Intermediate:** Projects a little more advanced with things like game development and very basic data analysis.
- **Advanced:** Competitive Programming, Advanced Algorithms, and Data Structures.

Comparison with our case:

- **Advantages:** Use of game-based learning from Algorithmics is a great tool to keep the child interested in his work.

- **Disadvantages:** Their focus on younger audiences might mean less depth in advanced topics. Our project targets a broader age range and provides a more thorough exploration.

Results and conclusions

Main results

The project includes the results in the development of educational materials for teaching Python programming, focusing on topics NumPy, strings, and lists.

The resulting products may be summarized as follows:

1. **NumPy**: includes all basics of numpy like array creation, array attributes, indexing, slicing, mathematical operation, etc. A second notebook on this topic was created to give the students some exercises and will roll in the same theme.
2. **Strings**: a notebook explaining about strings and how to use them in Python. It covers basic operations, slicing, in-built methods, regular expressions, and encoding/decoding, with practical exercises embedded. The Yandex Contest on this topic for interactive usage was also created.
3. **Lists**: a detailed note on lists properties and usage, slicing, built-in methods, multi-dimensional lists, stacks, queues, and iteration is well supported by coding exercises. The Yandex Contest on this topic for interactive usage was also prepared.

Prospects of further work

Although the project has now created a full suite of educational materials for Python programming, there remain several directions that future work could focus on:

1. Advanced topics and specialized modules:

- Create more modules on some advanced topics in Python such as Machine learning, Data visualization and Web development.

2. Improved advanced interactive features:

- Automate the checking of exercises feedback-taking personal info, grades etc.
- Get hands-on tutorial on interactive visualization tools & dashboards to consume data.

3. Multilingual support:

- Provide students with the opportunity to choose the language in studying the materials.

4. Evolution based on feedback and updates from Python.:

- Keep materials up to date with feedback from students.
- Revise the material according to new versions of python.

Conclusions

The main accomplishment is the creation of a complete set of educational materials for Python programming. The project provides well-known instructionals, interactive Jupyter Notebooks, and contests on Yandex Contests targeting to make the learning experience enjoyable.

The project achieved its aim. The emphasis in this project was on practical learning with hands-on exercise and coding. Potential future work includes scaling the contents by translating the materials to update them. The learning material created by the project represents a useful addition to the existing pool of teaching resources on Python programming.

List of sources

1. Python Software Foundation. Python Documentation:
<https://docs.python.org/>
2. Codim (Кодим):
<https://codim.ru/>
3. Yandex Lyceum (Яндекс Лицей):
<https://yandexlyceum.ru/>
4. Algorithmics (Алгоритмика):
<https://algorithmicschool.com/>
5. Jupyter Project. Project Jupyter:
<https://jupyter.org/>
6. String Documentation:
<https://docs.python.org/3/library/string.html>
7. Lists Documentation:
<https://docs.python.org/3/tutorial/datastructures.html>
8. NumPy Developers. NumPy Documentation:
<https://numpy.org/doc/>