INTERNATIONAL JOURNAL OF TECHNOLOGY AND ACADEMIC RESEARCH

ISSN: 2224-6878 | Pakistan Volume 6, No. 10.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

METHODOLOGICAL FOUNDATIONS FOR CREATING AN E-LEARNING RESOURCE FOR LEARNING THE PYTHON PROGRAMMING LANGUAGE

Termez University of Economics and Service
Master's Student: **Almardonova Shaxlo Soatmuradovna**
Academic Supervisor**: Namazov Gafur Shakulovich**
https://doi.org/10.5281/zenodo.23220186

Abstract: The article examines the pedagogical and methodological potential of using an electronic educational resource in the process of learning the Python programming language. The study emphasizes the importance of combining theoretical knowledge with practical tasks, organizing students' independent learning, mastering educational materials step by step, and providing timely assessment of learning outcomes. Based on these principles, the structure of an electronic educational resource designed for learning Python is proposed, and its content and functional capabilities are considered.

The proposed resource includes theoretical materials, programming code examples, practical and independent tasks, test questions, and mechanisms for assessing learning outcomes. It is argued that the use of an electronic educational resource in Python education can contribute to the development of algorithmic thinking, independent programming skills, students' learning motivation, and the effectiveness of mastering educational materials.

Keywords: Python, programming language, electronic educational resource, digital education, programming teaching methodology, independent learning, algorithmic thinking, interactive learning, practical tasks.

Introduction

The rapid development of information and communication technologies today necessitates the enhancement of traditional forms of organizing the educational process. Beyond serving merely as a means to deliver educational content, digital technologies are becoming vital pedagogical tools that enable students to engage in independent learning, monitor their knowledge, and develop practical skills. UNESCO's 2023 Global Education Monitoring Report highlights the significant role digital technologies play in delivering content, fostering skill development, and supporting the educational process. At the same time, the report emphasizes that the use of technology must be aligned with learners' needs and educational objectives. The use of electronic educational tools is particularly relevant in the teaching of programming, as learning to program extends far beyond the rote memorization of theoretical concepts. Students must master a set of interconnected skills, including algorithmic thinking, problem analysis, the development of solution algorithms, coding, error detection, and result analysis. The Python programming language stands out as a highly effective tool for this process. Official Python documentation highlights the language's simple and readable syntax, high-level data structures, and capabilities for rapid application development. The official Python "Beginner's Guide" also emphasizes the language's accessibility for newcomers. Therefore, creating a methodologically sound e-learning resource for learning Python is crucial for developing students' foundational programming knowledge and practical skills.

1. The primary objective of the research is to develop the content, structure, and functional capabilities of a methodologically grounded e-learning resource for learning Python, and to propose a methodology for its use. To achieve this objective, the following tasks were

ISSN: 2224-6878 | Pakistan Volume 6, No. 10.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

defined: To analyze existing methodological approaches to teaching the Python programming language; to identify the main difficulties students encounter when learning programming; to examine the pedagogical potential of an electronic educational resource; to develop the content structure of an electronic resource for learning Python; to integrate theoretical and practical materials into a unified electronic environment; to design a system of assignments that supports students' independent learning; and to propose testing and assessment mechanisms to evaluate knowledge and skills.

2. Research Methodology: The research process may employ methods such as systems analysis, comparative analysis, pedagogical observation, modeling, generalization, and design.

When creating the electronic resource, it is advisable to adopt the "theory — example — practice — assessment — analysis" principle. Under this approach, each topic is introduced with a brief theoretical explanation, followed by a sample of program code, and then a practical assignment for the student to complete independently. Finally, the level of mastery is determined through a test or assessment task. This approach ensures that the student becomes an active participant in the learning process rather than remaining a passive listener.

3. The importance of electronic resources in teaching the Python programming language. Using electronic resources during the process of learning Python creates a range of pedagogical opportunities. First, the learning material is systematized; students master topics in a specific, logical sequence. Second, electronic resources expand opportunities for independent study. Students can review theoretical material, study code samples, and complete assignments without the direct involvement of an instructor. Third, electronic resources enable rapid assessment of knowledge. Tests, practical assignments, and review questions help determine the student's level of mastery. Fourth, electronic resources allow for the use of multimedia elements. For instance, images explaining code execution, flowcharts, video lessons, and interactive examples enhance the clarity and comprehensibility of the learning UNESCO emphasizes that when using educational technologies, what matters is not the technology itself, but rather how well it aligns with educational goals. Therefore, an electronic resource should be designed not merely as a collection of data, but as a learning environment with a specific pedagogical purpose. Proposed structure of the electronic resource:

It is proposed that the electronic resource designed for learning the Python programming language consist of the following modules:

Module 1. Introduction to the Python programming language

This module covers the origins, evolution, key features, and application areas of the Python programming language, as well as how to set up the programming environment.

Module 2. Basic Python syntax

Variables, constants, identifiers, comments, operators, and the general structure of a Python program are explained.

Module 3. Data types

Numeric data, strings, boolean values, and other fundamental data types are taught using examples.

Module 4. Operators

The functions of arithmetic, comparison, logical, and other operators are explained through practical examples.

Module 5. Conditional statements

Programming conditional logic using `if`, `elif`, and `else` statements is taught.

Module 6. Iterative processes

Exercises on programming iterative algorithms using `for` and `while` loops are provided.

ISSN: 2224-6878 | Pakistan Volume 6, No. 10.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

Module 7. Functions

The concept of functions, parameters, arguments, return values, and the use of functions are taught.

Module 8. Data structures

Practical tasks involving lists, tuples, dictionaries, and sets are included.

Module 9. Working with files

Exercises are developed for creating, reading, and writing files, as well as processing data within them.

Module 10. Object-Oriented Programming

Key concepts such as classes, objects, attributes, methods, and inheritance are covered.

Module 11. Practical projects

Students develop small software projects using the knowledge they have acquired.

Module 12. Testing and final assessment

Theoretical material → Code sample → Explanation → Practical task → Independent work → Test → Result assessment.

The advantage of this model is that the student not only reads about new knowledge but also tests it in practice. Theoretical knowledge is transformed into practical skills as the student writes the program code and observes the results. For example, regarding the topic of "Conditional Operators," the functions of `if`, `elif`, and `else` operators are explained first. Then, a simple program example is presented. Following this, the student is required to independently program a solution to a given problem. Finally, the level of understanding is assessed using test questions related to the topic. This methodological sequence ensures the integration of theory and practice, which is crucial for learning programming. Functional capabilities of the electronic resource: It is advisable for the electronic resource to possess the following functional capabilities: a user-friendly and simple interface; the ability to study topics sequentially; viewing theoretical materials; accessing sample program codes; completing practical tasks; answering test questions; automatic or semi-automatic assessment of results; viewing learning outcomes; using supplementary materials for independent study; and accessing an electronic dictionary and terminology database. Future development of the resource could incorporate features such as an interactive coding environment, automated code checking, user profiles, learning progress statistics, and an adaptive task system. The expected outcomes involve fostering the following competencies in students through the use of the e-learning resource: understanding fundamental concepts of the Python programming language; algorithmic and logical thinking; formulating problems as algorithms; writing program code independently; identifying and debugging errors; utilizing functions and data structures; developing small-scale software projects; and acquiring independent learning skills. Experimental trials can be conducted to evaluate the effectiveness of the electronic resource. A suitable approach involves dividing students into experimental and control groups and comparing the results of initial and final assessments. For instance, the experimental group would study Python topics using the developed electronic resource, while the control group would be taught using traditional methods. At the conclusion of the course, a comparative analysis would be performed on test results, the completion rate of practical assignments, and indicators of independent work. This process would enable the formulation of a scientifically grounded conclusion regarding the educational effectiveness of the electronic resource.

Conclusion: Creating an e-learning resource for learning the Python programming language represents a promising avenue for enhancing the effectiveness of programming instruction within the modern educational process. The study established the rationale for integrating theoretical

ISSN: 2224-6878 | Pakistan Volume 6, No. 10.

IF: 14.91 | 2026 http://waerasia.org/index.php/ijtar

materials, practical code examples, independent assignments, tests, and assessment mechanisms into a unified system when developing the resource. The proposed model—comprising theory, code examples, practical tasks, independent work, testing, and assessment—facilitates the step-by-step development of students' knowledge. This approach enables not only the mastery of Python's core constructs but also the development of skills in algorithmic thinking, the analysis of problem situations, and the creation of software solutions. Future stages of the research will involve developing a practical prototype of the resource, conducting pilot testing, and evaluating its effectiveness based on statistical criteria. In this regard, the approach also serves as a foundation for the practical component of a master's thesis.

References

1. Python Software Foundation. The Python Tutorial. Python 3 Documentation. Python.org.
2. Python Software Foundation. Python for Beginners. Python.org.
3. Python Software Foundation. About Python. Python.org.
4. UNESCO. Global Education Monitoring Report 2023: Technology in education – A tool on whose terms? Paris: UNESCO, 2023.
5. Alammary, A. Blended learning models for introductory programming courses: A systematic review. PLOS ONE, 2019.
6. Scherer, R., Siddiq, F., Sánchez Viveros, B. A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. Computers in Human Behavior, 2020, Vol. 109, 106349.
7. Calderon, I., Silva, W., Feitosa, E. Active Learning Methodologies for Teaching Programming in Undergraduate Courses: A Systematic Mapping Study. Informatics in Education, 2024, Vol. 23(2), pp. 279–322.
8. Jiménez-Toledo, J. A., Collazos, C., Revelo-Sánchez, O. Considerations for the Teaching-Learning Processes of Introductory Programming Courses: A Systematic Literature Review. TecnoLógicas.
9. Rovshenov, A., Sarsar, F. Research trends in programming education: A systematic review of the articles published between 2012–2020. Journal of Educational Technology and Online Learning, 2023, Vol. 6(1).
10. Katona, J., Katonane Gyonyoru, K. I. Enhancing programming education through AI-driven tools and innovative pedagogical models: a systematic study on student engagement, retention, and self-regulation. Journal of Computers in Education, 2026.