




The effect of python programming language teaching on 7th grade students' programming self-efficacy skills

Feryad Doğuş ¹, Yasin Özkan ², Sibel Barin Özkan ²

¹ Kırılı Secondary School, Ministry of National Education, Ordu, Turkey; ² Department of Computer Technologies, Zonguldak Bulent Ecevit University, Zonguldak, Turkey

ARTICLE HISTORY

Received: 01.07.2024

Accepted: 20.09.2024

KEYWORDS

Python

Programming education

Programming self-efficacy

CORRESPONDENCE

Feryad Doğuş, feryaddogus@gmail.com

ABSTRACT

This study aimed to investigate the impact of teaching the Python programming language on the programming self-efficacy of 7th-grade students. A quantitative research design was employed, utilizing a one-group pretest-posttest model. The study sample consisted of 10 students enrolled in the 7th grade at a public school located in a rural district of northern Turkey during the spring semester of the 2023-2024 academic year. These students attended a school affiliated with the Ministry of National Education. The study group was selected using the convenience sampling method, a type of purposive sampling. Data for this research was collected using the Programming Self-Efficacy Scale for Secondary School Students, developed by Kukul, Gökçearslan, and Günbatar (2017). The collected data were analyzed using the Jamovi software. To assess the suitability of the data for normal distribution, skewness and kurtosis coefficients, as well as Shapiro-Wilk test values (a normality test), were examined, alongside an inspection of histogram plots. The results indicated that the data were normally distributed with respect to both research questions. For the first research question, a paired sample t-test was conducted to determine whether there was a statistically significant difference between the pre-test and post-test scores. The analysis revealed that Python instruction did not have a significant effect on the programming self-efficacy of the 7th-grade students. Regarding the second research question, the homogeneity of the data was assessed. Levene's test for equality of variances was performed, confirming that the assumption of homogeneity was met. Consequently, independent sample t-tests were conducted to examine the significance of the difference in pre-test and post-test scores based on gender. The results indicated no statistically significant difference in programming self-efficacy scores between male and female students.

Introduction

With the rapid advancement of technology and increasing digitalization, programming education, which is among the 21st century skills, has become more important. Programming skills are a

comprehensive tool not only in information technologies but also in students' skills such as analytical thinking, problem solving, creative thinking and logical inference (Grover & Pea, 2013). These skills, which students acquire at an early age, contribute to their success not only in their educational processes but also in their future professional lives (Shute et al., 2017).

Programming education given in secondary schools directly affects students' cognitive development processes. In the literature, it is stated that programming education received by students in this age group improves their cognitive flexibility, algorithmic thinking skills and problem-solving strategies (Futschek, 2006). In addition, programming education also contributes to emotional and social learning dimensions, increases students' self-confidence and reinforces their motivation for learning processes (Sáez-López et al., 2016).

In this context, Python programming language is a frequently preferred language used for educational purposes. Python programming language is easy to learn, has a simple syntax and offers a wide usage area. For this reason, it is seen as a preferred learning tool by both educators and students. The fact that it is understandable and applicable for students who are new to programming education makes Python programming language a language used in secondary schools (Lye & Koh, 2014). In addition, Python's modular structure and extensive libraries enable students to work on different projects and enrich their learning processes (Guzdial, 2015).

According to the research, Python programming language improves students' programming self-efficacy skills. Self-efficacy is the belief that an individual can successfully fulfill a certain task. This belief is one of the main sources of motivation in learning processes (Bandura, 1997). Students with high levels of self-efficacy are more resilient in coping with the difficulties they face. These students are also determined to achieve their goals (Schunk & Pajares, 2009). In this context, Python programming language positively affects not only students' technical skills but also their overall educational performance.

The main purpose of this study is to investigate the effect of Python programming language on 7th grade students' programming self-efficacy skills. In literature, there are various studies on the positive contributions of providing programming education to students at an early age to their cognitive and emotional development (Papert, 1980; Wing, 2006). However, there are almost no studies examining the effects of Python programming language on students especially at the 7th grade level. This study aims to fill this gap in the literature and examine the potential benefits of Python programming language in education in more depth. This study, which will provide important information for educators, aims to reveal the effects of programming education on student achievement. In line with this purpose, the following questions were sought to be answered in the study:

1. Does Python instruction have an effect on 7th grade students' programming self-efficacy skills?
2. Do students' programming self-efficacy skills differ according to gender?

Theoretical framework

Theoretical foundations of using python in education

Constructivist Learning Theory: According to constructivist learning theory, students should actively construct knowledge (Piaget, 1972). Python programming language contributes to this theory by giving students the opportunity to write their own programs. Students experience an active learning process by writing code and solving problems. Research examining the function of Python in education in the context of constructivist learning theory supports students' active construction of knowledge through coding. For example, Grover and Pea (2013) reveal that in a

constructivist learning environment, students gain experience while writing programs and actively work to make sense of the knowledge they learn. Similarly, Lye and Koh (2014) emphasize that Python programming language education fosters constructivist learning and provides students with problem solving and critical thinking skills.

Sequential Development and Problem Solving: According to Vygotsky's (1978) Social Development Theory, learning is a social process, and students learn by interacting with more experienced individuals. The Python programming language supports students to come together and learn together through online communities and collaborative projects. Research conducted within the framework of Vygotsky's Social Development Theory has shown that the Python programming language supports students' learning through social interaction. For example, Kafai and Burke (2015) found that group projects using Python fostered social learning among students and thus developed a more meaningful understanding of problem solving.

Information Processing Theory: According to information processing theory, the learning process is defined as encoding, storing and retrieving information (Atkinson & Shiffrin, 1968). The Python programming language enables students to actively process information by writing code and debugging the program. Coding strengthens students' analytical thinking and problem-solving skills. Studies investigating the impact of Python within the scope of information processing theory have shown that the coding process improves students' information processing skills. For example, Luxton-Reilly (2016) emphasizes that Python programming language enables students to actively apply cognitive processes such as encoding, storing and retrieving information during the learning process.

Using python in education

Python programming language is used in secondary school education to support students' analytical thinking, problem solving and algorithmic thinking skills. For example, basic calculation applications can be done with Python in math lessons. Students learn to perform automatic calculations using loops and conditional statements. In science, Python programming language can be used for simple simulations and data analysis, where students can learn how to collect data, analyze it and interpret the results. Python's graphical libraries can also be used to create visual programming applications, giving students opportunities to design games or develop interactive stories. Such projects allow students to develop both their coding skills and their creative thinking skills.

Python in Beginner Programming Education: The use of Python in beginner programming education is related to its simplicity and easy readability. Python has less complex syntax than other programming languages, which makes it easier for students to learn basic programming terms.

STEM Education and Python: The use of Python programming language in STEM (Science, Technology, Engineering, Mathematics) education is becoming increasingly common. Python's extensive library support allows students to practice in a variety of areas such as data analysis, scientific calculations and engineering applications.

Robotics and Python: Robotic coding education is one of the most effective tools to provide students with engineering and programming skills. Python programming language is widely used in robotic coding and provides an opportunity for students to gain experience writing code with real-world applications.

Data Science and Python: Data science education develops skills to analyze and interpret large data sets. Python's data processing and visualization libraries (such as pandas, matplotlib, seaborn) enable students to acquire these skills.

The use of Python programming language in education is supported by various theoretical foundations and research in the literature. Theories such as constructivist learning approach, social development theory, and information processing make Python programming language an effective tool in education. Research in the literature reveals that Python is an effective educational tool in a wide range of areas from beginner programming to STEM education, from robotic coding to data science.

Method

Research model

The aim of the study is to examine the effect of teaching Python programming language on 7th grade students' programming self-efficacy skills. In order to realize this aim, a single group pretest-posttest design was adopted, and a quantitative research method was applied. Experimental designs are research methods used to reveal cause-and-effect relationships between variables. This type of research is designed to examine the effects of at least one independent variable on one or more dependent variables. In this study, after the Python programming language, which is the independent variable, was taught to the students, pre- and post-experiment measurements were made.

Study group

Ten seventh-grade middle school students from a public school connected to the Ministry of National Education make up the study's sample. The student group consisted of 12- and 13-year-old students, 3 boys and 7 girls. Convenient sampling method, one of the purposive sampling methods, was used in the selection of the students. The most critical factor in using the convenience sampling method is that the sample should be easily accessible, suitable and convenient for the study (McMillan & Schumacher, 2014). In this study, the fact that the researcher was a teacher and taught at the same school made it easy to reach the sample group.

Data collection tool

In the study, the "Programming Self-Efficacy Scale for Middle School Students" created by Günbatar, Gökçearslan and Kukul (2017) was used to evaluate the potential effect of Python programming language on 7th grade students' programming self-efficacy skills. There are a total of 31 items on the scale. Each item is in the form of a five-point Likert scale. The loadings of the items of the scale ranged between 0.47 and 0.71 and explained 41.15% of the total variance. In the analyses evaluating internal consistency, Cronbach's alpha coefficient was found to be 0.95 and the two-half test reliability result was 0.96, indicating that the scale has a high internal consistency. Exploratory and confirmatory factor analyses conducted to examine construct validity confirmed that the scale was valid and reliable.

Implementation process

The applications were completed over the course of an eight-week course period in a research project involving seventh-grade students at a public school connected to the Ministry of National Education during the second semester of the 2023–2024 academic year. During this training period, students were given basic Python training. Each student was given the opportunity to make one-to-one applications on computers with very good technological infrastructure. These

applications were carried out in the coding classroom of the school under the control of the course teacher using the show-and-do technique. The students were allowed to enter the Google Meet application by the teacher and then the relevant topic of each week was shown to the students by the teacher in the Python 3.10 application. Then, these applications were expected to be realized by the students. The training process is presented in detail week by week in Table 1:

Table 1 Basic Python training process

Week	Education
1. Week	Install Python 3 on Windows
2. Week	Math operations in Python
3. Week	Strings
4. Week	Print function
5. Week	String process
6. Week	Variables
7. Week	Lists
8. Week	Input function and format method

Data analysis

Whether the data fit the normal distribution was determined by looking at the skewness and kurtosis coefficients, Shapiro-Wilk test values and histogram graphs in the analysis using the Jamovi program. After establishing that the data were normally distributed, the difference between the pre- and post-test scores was examined using the dependent sample t-test.

Results

The effect of python instruction on 7th grade students' programming self-efficacy skills

In line with the first problem of the study, the following hypotheses were tested.

H0: Python instruction has no effect on 7th grade students' programming self-efficacy skills.

H1: Python instruction has an effect on 7th grade students' programming self-efficacy skills.

Table 2 displays the findings of the examination of the normality tests conducted on the pre-test and post-test difference scores of the Python instruction aimed at enhancing the students' programming self-efficacy skills in the seventh grade.

Table 2 Descriptives

	N	Mean	Median	Mode	Sd	Skewness	Kurtosis	p
Difference	10	2.10	5.50	-34.0	21.6	-0.560	-0.840	0.365

As seen in Table 1, when the normality level is examined according to the Shapiro-Wilk test, it is seen that the distribution of the difference score between the programming self-efficacy pretest and posttest scores is normal ($p > 0.05$). In addition, the skewness and kurtosis values are between -1 and +1, and the histograms, box plots and Q-Q Plots graphs show that the data of the difference score show a completely normal distribution. A dependent sample t-test was conducted to reveal the significance of this difference. Table 3 displays the dependent sample t test findings.

Table 3 Paired Samples T-Test

			Statistic	df	p
Self-Efficacy Pre-Test	Self-Efficacy Post-Test	Student's t	-0.307	9.00	0.766

When Table 3 is examined, it is observed that there is no significant difference between 7th grade students' programming self-efficacy pre-test and post-test scores [$t(9)=-0.307$, $p > 0.05$]. The study's null hypothesis (H_0), "Python instruction has no effect on 7th grade students' programming self-efficacy skills," was accepted in line with this finding.

2. Investigation of Students' Programming Self-Efficacy Skills by Gender

In line with the second problem of the study, the following hypotheses were tested.

H_0 : Students' programming self-efficacy skills do not differ according to gender.

H_1 : Students' programming self-efficacy skills differ according to gender.

2.1. Findings Regarding the "Gender" Status of Pre-Test Scores

The results of the analysis of the normality tests of the pre-test scores of the students' programming self-efficacy skills are shown in Table 4.

Table 4 Descriptives

	N	Mean	Median	Mode	Sd	Skewness	Kurtosis	p
Pre-test	10	102	103	107	9.50	0.0471	0.243	0.724

It shows normal distribution at $p > 0.05$ level.

As seen in Table 4, when the normality level is examined according to the Shapiro-Wilk test, it is seen that the distribution of programming self-efficacy pretest scores is normal ($p > 0.05$). The skewness and kurtosis values are between -1 and +1 and the histogram, box plot and Q-Q plot graphs show that the data belonging to the scores exhibit a completely normal distribution. In this case, the homogeneity of variances was tested to decide which test to conduct. In this direction, Levene's test was conducted. The results of Levene's test are presented in Table 5. According to Levene's test, homogeneity of variances was achieved ($p=0.135 > 0.05$).

Table 5 Homogeneity of Variances Tests

		F	df	df2	p
Self-Efficacy Pre-Test	Levene's	2.77	1	8	0.135

Accordingly, an independent sample t test was conducted to reveal the significance of the pre-test scores according to the gender variable. The results of the independent sample t test are shown in Table 6.

Table 6 Independent Samples T-Test

		Statistic	df	p
Self-Efficacy Pre-Test	Student's t	0.820	8.00	0.436

When Table 6 is examined; students' programming self-efficacy pretest scores do not show a significant difference according to gender [$t(8)=0.820$, $p > 0.05$].

2.2. Findings Related to the "Gender" Status of Posttest Mean Scores

The results of the analysis of the normality tests of the post-test scores of the students' programming self-efficacy skills are shown in Table 7.

Table 7 Descriptives

	N	Mean	Median	Mode	Sd	Skewness	Kurtosis	p
Son Test	10	104	112	115	17.1	-1.03	-0.217	0.054

It shows normal distribution at $p > 0.05$ level.

As seen in Table 7, when the normality level is examined according to the Shapiro-Wilk test, it is seen that the distribution of programming self-efficacy post-test scores is normal ($p > 0.05$). It is seen that the skewness and kurtosis values are approximately between -1 and +1 and the histogram, box plot and Q-Q plot graphs show that the data belonging to the scores exhibit a completely normal distribution. In this case, the homogeneity of variances was tested to decide which test to conduct. In this direction, Levene's test was conducted. The results of Levene's test are presented in Table 8.

Table 8. Homogeneity of Variances Tests

		F	df	df2	p
Self-Efficacy Posttest	Levene's	0.00422	1	8	0.950

According to Levene's test, homogeneity of variances was achieved ($p=0.950 > 0.05$). Accordingly, an independent sample t test was conducted to reveal the significance of the post-test scores according to the gender variable. The results of the independent sample t test are shown in Table 9.

Table 9. Independent Samples T-Test

		Statistic	df	p
Self-Efficacy Posttest	Student's t	-0.699	8.00	0.504

When Table 9 is examined; students' programming self-efficacy posttest scores do not show a significant difference according to gender [$t(8)=-0.699$, $p > 0.05$]. In the context of the findings of the second problem, the null hypothesis (H_0) "Students' programming self-efficacy skills do not differ according to gender" was accepted.

Conclusion and discussion

In this study, the effect of Python programming language on 7th grade students' programming self-efficacy skills was examined. In the context of the findings of the study, in general, Python programming education did not make a statistically significant difference on students' self-efficacy skills. The fact that the difference between the pre-test and post-test results was not significant indicates that Python programming language instruction did not have the expected effect. This shows that Python programming language instruction has a limited effect on increasing students' programming self-efficacy skills. However, in the interpretation of these results, the short duration of the training and the students' prior knowledge of the programming language should be taken into consideration (Bandura, 1997).

The findings obtained in the study do not show a significant difference when analyzed in terms of gender variable. There is no statistically significant difference in programming language self-efficacy levels between male and female students. This result is in line with the studies in the literature which indicate that the effect of gender on programming skills is limited. However, it is suggested that the indirect effects of gender variable should be examined in depth by considering factors such as gender roles and prejudices towards STEM fields.

The results obtained reveal that teaching Python programming language should be supported by constructivist teaching strategies in order to be effective. The constructivist approach supports students' active construction of knowledge and their participation in problem-solving processes (Piaget, 1972; Vygotsky, 1978). In this context, it should be emphasized that Python programming

language is an important tool not only for students' technical skills but also for developing 21st century skills such as creative thinking and problem-solving skills. Activities such as projects and group work that will increase the active participation of students should be included more in this process.

In the context of the findings of the study, it is seen that Python programming language training did not have an effect on students' self-efficacy skills at the expected level. However, it should be taken into consideration that students' interest in programming and their prior knowledge levels may have an impact on these results. In the literature, it is emphasized that self-efficacy skills are affected by individuals past success levels and feedback in learning processes (Bandura, 1997). For this reason, providing students with more practice opportunities and reinforcing the sense of success are very important for the development of self-efficacy skills.

In line with the findings of this study, it is recommended that Python programming language training should be handled in a longer and more comprehensive manner than an 8-week period. In particular, it is thought that training strategies tailored to the individual needs of students may be more effective in developing self-efficacy skills. Future research may examine the effects of teaching Python programming language in different age groups and at different educational levels from a broader perspective. The small number of students in this study is also a limitation of the study and it is recommended that future studies should include more students in Python programming language training.

Disclosure statement

No potential conflict of interest was reported by the author(s).

References

- Atkinson, R. C., & Shiffrin, R. M. (1968). Human memory: A proposed system and its control processes. In K. W. Spence & J. T. Spence (Eds.), *The psychology of learning and motivation*, (2), pp. 89-195. Academic Press. [https://doi.org/10.1016/S0079-7421\(08\)60422-3](https://doi.org/10.1016/S0079-7421(08)60422-3)
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. W.H. Freeman and Company.
- Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. In *International Conference on Informatics in Secondary Schools – Evolution and Perspectives* (pp. 159-168). Springer. https://doi.org/10.1007/11915355_15
- Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43. <https://doi.org/10.3102/0013189X12463051>
- Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6), 1-165. <https://doi.org/10.2200/S00684ED1V01Y201511HCI033>
- Kafai, Y. B., & Burke, Q. (2015). *Connected code: Why children need to learn programming*. MIT Press. <https://doi.org/10.7551/mitpress/9992.001.0001>
- Kukul, V., Gökçearslan, Ş., & Günbatar, M. S. (2017). Programming self-efficacy scale for middle school students: Development, validity, and reliability. *Educational Technology Theory and Practice*, 7(1), 158-179.
- Luxton-Reilly, A. (2016). Learning to program is easy. *Proceedings of the 19th Australasian Computing Education Conference (ACE 2016)*, 160, 284-293. <https://dl.acm.org/doi/10.1145/2899415.2899432>
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61. <https://doi.org/10.1016/j.chb.2014.09.012>

- McMillan, J. H., & Schumacher, S. (2014). *Research in education: Evidence-based inquiry* (7th ed.). Pearson Higher Ed.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Piaget, J. (1972). *The principles of genetic epistemology*. Routledge.
- Piaget, J. (1972). *The psychology of the child*. Basic Books.
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using “Scratch” in five schools. *Computers & Education*, 97, 129-141. <https://doi.org/10.1016/j.compedu.2016.03.003>
- Schunk, D. H., & Pajares, F. (2009). Self-efficacy theory. In K. R. Wentzel & A. Wigfield (Eds.), *Handbook of motivation at school* (pp. 35-53). Routledge.
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press. <https://doi.org/10.2307/j.ctvjf9vz4>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.111821>