

The Impact of Python Usage in the Teaching of Physics

Lorena Kelo^{1*}, Eglantina Merkaç², and Idajet Selmani³

¹Department of Mathematics and Physics, Faculty of Natural and Human Science, "Fan S. Noli" University, Korça, Albania

²Department of Engineering, Faculty of Applied and Economical Science, Albanian University, Tirana, Albania

³Department of Physics, Faculty of Natural Science, "Aleksander Xhuvani" University, Elbasan, Albania

Abstract. The integration of programming tools in science education has reshaped the teaching and learning of Physics, offering new opportunities for conceptual understanding and student engagement. This study examines the impact of using Python, together with Jupyter Notebooks and GitHub, in teaching introductory Physics courses to first-year Computer Engineering and Information Technology students. A quasi-experimental design compared one semester of instruction integrating Python with a subsequent semester delivered without programming tools. Quantitative analysis (SPSS) of midterm and final exam results revealed statistically significant improvements in students' performance when Python was applied ($p < 0.01$), while qualitative reflections highlighted enhanced conceptual understanding, increased motivation, and the development of algorithmic and visual thinking skills. Although no significant differences were found in attendance, students reported greater interaction, participation, and independent exploration. The findings suggest that Python-based teaching environments not only improve academic outcomes but also foster active, student-centered learning, making Physics more accessible and relevant within technology-oriented curricula.

1 Introduction

Over the past few decades, the teaching of Physics has undergone significant transformation due to the growing integration of digital technologies in higher education. Among these developments, the incorporation of programming languages as instructional tools has gained increasing attention. Python, in particular, has emerged as both a versatile programming language and an effective pedagogical resource, enabling students to model, simulate, and visualize physical phenomena in ways that enhance conceptual understanding. As emphasized by Sherin et al. [1], computer technology in physics education should serve as a means of promoting conceptual understanding through interaction with models and

* Corresponding author: lkelo@unkorce.edu.al

simulations, rather than as an end in itself. Computational tools allow students to engage directly with physical systems, encouraging exploration and deeper reasoning [2]. Supported by libraries such as SymPy, SciPy, and VPython, Python provides an accessible environment for integrating modeling and simulation into the learning process.

A substantial body of research [3–8] indicates that students involved in computational modeling activities demonstrate improvements in both conceptual and procedural understanding. Moreover, embedding programming within science education aligns classroom practice with authentic scientific inquiry and strengthens students' role as active participants in the learning process [9].

In the Albanian context, Basholli et al. [10] reported that the use of Python enhanced student engagement and facilitated the comprehension of complex concepts through interactive and project-based activities.

Interactive platforms such as Jupyter Notebooks further extend these advantages by supporting real-time experimentation and immediate feedback [11]. The integration of executable code enables students to modify parameters and observe outcomes dynamically, thereby strengthening algorithmic thinking and computational skills [12, 13]. Additionally, dynamic visualizations support stronger connections between empirical observation and theoretical modeling [14].

These pedagogical approaches are grounded in constructivist learning theory. Papert [15] emphasized that programming supports deep learning by framing knowledge construction as an active and creative process. In this perspective, technology functions not merely as a medium for information delivery, but as a cognitive tool that facilitates meaningful engagement with disciplinary concepts.

The present study investigates the impact of integrating Python into introductory Physics courses, examining its effects on students' conceptual understanding, motivation, and engagement. Through empirical analysis, it seeks to determine whether Python-based instruction contributes to improved learning outcomes.

Ultimately, the integration of Python into Physics education represents not only a methodological enhancement but also a pedagogical approach aligned with the demands of technology-oriented curricula and contemporary scientific practice.

2 Methodology

The study followed the same cohort of 45 first-year students enrolled in the Bachelor's programs in Computer Engineering and Information Technology across two consecutive semesters. During the first semester (Physics 1), Python, together with Jupyter Notebooks and GitHub, was integrated into the instructional process to support simulation, modeling, and interactive exploration of physical phenomena [12, 14]. In the second semester (Physics 2), the same group of students continued their studies without the integration of programming tools, following a more traditional instructional approach.

This design enabled a comparison of student performance under two instructional conditions (with and without Python integration). For analytical purposes, each semester was treated as a distinct instructional condition. Accordingly, the results from the two semesters were analysed as separate sets of observations, resulting in a total dataset of 90 observations (45 students \times 2 semesters). This approach enabled the examination of whether observed differences in academic performance were associated with the instructional method, while maintaining the same student cohort across both conditions.

2.1 Tools and Resources

The instructional approach incorporated the following tools:

- Jupyter Notebooks, used as an interactive environment for real-time modeling, experimentation, and instructional delivery.
- GitHub, utilized for distributing materials, facilitating remote collaboration, and documenting code and project development[†].
- Python, employed as the primary programming language together with its scientific libraries.

2.2 Structure of Teaching

The instructional design combined theoretical explanation with hands-on computational activities:

- Guided exercises delivered through Jupyter Notebooks containing open-source code shared via GitHub, where students modified parameters and analysed results in real time [11, 13].
- Laboratory sessions in which students used Python to simulate physical phenomena such as motion and forces [8, 16, 17].
- Small-scale individual projects aimed at encouraging exploration, independent application of concepts, and problem-solving through coding.

2.3 Data Collection

Data were collected through multiple sources to ensure both quantitative and qualitative insight:

- Analysis of midterm and final examination results from both semesters.
- Direct observation of student participation and classroom interaction.
- Assessment of students' prior background in Physics and basic programming knowledge.
- Student reflections regarding their learning experience following the implementation of the Python-based approach.

Reflections were collected from 41 students who were present at the end of the semester. The qualitative analysis followed a descriptive approach based on identifying recurring patterns and representative statements rather than applying a formal thematic coding procedure.

2.4 Data Analysis

Statistical analysis was conducted using SPSS software to identify statistically significant differences between instructional methods and to determine whether observed performance variations could be attributed to the integration of programming tools. In addition, qualitative analysis of student comments and reflections was performed to capture perceived benefits and subjective learning experiences.

2.5 Limitations

The study presents several limitations. These include the relatively small sample size, variations in students' prior programming knowledge, and dependence on technological infrastructure, such as computer availability, internet access, and proper configuration of the Jupyter environment. These factors may influence the generalizability of the findings.

[†] <https://github.com/POSS-UniMe/simple-physics-with-Python>

3 Results and Discussions

This section presents the analysis of the impact of integrating Python, Jupyter Notebooks, and GitHub into Physics instruction during the first semester, compared to the second semester in which programming tools were not used. Statistical analyses were conducted using SPSS software. The statistical dataset comprised 90 observations derived from the two instructional conditions implemented across consecutive semesters.

The dataset includes the following variables: gender (“Gender”), duration of Physics studies in high school (“Background”), high school Physics average grade (“HS_Phys_Avg”), attendance percentage (“Attendance_Percentage”), midterm examination results (“Midterm”), and final course grade (“Final_Grade”). The variable “MethodUsed” distinguishes between the two instructional approaches and serves as the primary factor for evaluating the effect of programming integration on academic performance.

The analysis was conducted both comparatively and correlationally to identify potential relationships among the selected variables. Normality testing using the Shapiro–Wilk test (Table 1) yielded p-values below 0.05 for all variables, indicating non-normal distribution. Consequently, non-parametric statistical tests were applied.

Table 1. Tests of normality for the defined variables

	Shapiro-Wilk		
	Statistic	df	Sig.
HS_Phys_Avg	.887	90	.000
Attendance_percentage	.938	90	.000
Midterm	.958	90	.006
Final_Grade	.824	90	.000

The Mann–Whitney U test (Table 2) was performed to determine whether the instructional method influenced attendance, midterm performance, and final grades. The results show statistically significant differences for both “Midterm” ($p = 0.010$) and “Final_Grade” ($p < 0.001$), indicating that the Python-based approach had a measurable positive impact on academic performance. In contrast, the difference in attendance (“Attendance_Percentage”) was not statistically significant ($p = 0.077$), suggesting that the instructional method did not substantially affect students’ class attendance.

Table 2. Results for Mann-Whitney U Test

	Attendance_percentage	Midterm	Final_Grade
Mann-Whitney U	794.500	692.500	553.000
Wilcoxon W	1829.500	1727.500	1588.000
Z	-1.768	-2.584	-3.890
Asymp. Sig. (2-tailed)	.077	.010	.000

To further examine the influence of additional factors on final academic performance, Spearman’s rho correlation analysis was conducted (Table 3). The analysis revealed statistically significant positive correlations between “Final_Grade” and all three examined variables: “Background,” “HS_Phys_Avg,” and “Attendance_Percentage.”

Table 3. Correlations between variable pairs

			Final_Grade	Background	HS_Phys_Avg	Attendance_percentage
Spearman's rho	Final_Grade	Correlation Coefficient	1.000	.242*	.477**	.621**
		Sig.	.	.022	.000	.000
		N	90	90	90	90

The strongest correlation was observed between final grade and attendance ($\rho = 0.621$, $p < 0.001$), indicating that students with higher attendance rates tended to achieve better academic results. A moderate positive correlation was identified between final grade and high school Physics average ($\rho = 0.477$, $p < 0.001$). The duration of prior Physics studies (“Background”) showed a weaker but still statistically significant positive correlation ($\rho = 0.242$, $p = 0.022$). These findings suggest that academic success in university-level Physics is influenced by both prior preparation and sustained engagement throughout the course.

To further assess the combined predictive power of the variables, a linear regression model was constructed with “Final_Grade” as the dependent variable and “MethodUsed,” “HS_Phys_Avg,” “Attendance_Percentage,” “Gender,” and “Background” as independent variables.

Although the original data were not normally distributed, the standardized residuals of the regression model were tested for normality (Table 4).

It is important to note that linear regression requires normality of residuals rather than of the original variables. Since the standardized residuals did not significantly deviate from normality ($p = 0.538$), the assumptions required for linear regression were considered sufficiently satisfied.

Table 4. Test of normality for residuals

	Shapiro-Wilk		
	Statistic	df	Sig.
ZRE_1 Standardized Residual	.987	90	.538

The model summary (Table 5) shows an R^2 value of 0.622, indicating that approximately 62.2% of the variance in final grades is explained by the included predictors. This suggests a relatively strong explanatory capacity. The Durbin–Watson statistic (1.155) indicates slight positive autocorrelation among residuals; however, given the modest sample size, this value remains within an acceptable range.

Table 5. Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Durbin-Watson
1	.789	.622	.600	.996	1.155

The ANOVA results (Table 6) confirm that the regression model as a whole is statistically significant ($p < 0.001$), demonstrating that the combination of predictor variables reliably explains variation in students’ final grades.

Table 6. ANOVA

Model	Sum of Squares	df	Mean Square	F	Sig.
-------	----------------	----	-------------	---	------

1	Regression	137.232	5	27.446	27.688	.000
	Residual	83.268	84	.991		
	Total	220.500	89			

The regression coefficients (Table 7) indicate that the instructional method (“MethodUsed”) had a statistically significant and positive effect on final grades (B = 0.727, p = 0.001). In addition, both high school Physics average (B = 0.586, p < 0.001) and attendance (B = 0.031, p < 0.001) emerged as strong predictors of academic success.

Table 7. Coefficients of the regression model

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-2.918	.814		-3.585	.001
	Gender	.407	.284	.104	1.433	.156
	Background	.465	.234	.149	1.984	.051
	HS Phys Avg	.586	.121	.414	4.838	.000
	Attendance percentage	.031	.007	.337	4.165	.000
	MethodUsed	.727	.217	.232	3.346	.001

Gender was not a statistically significant predictor (p = 0.156), while the background variable approached significance (p = 0.051), suggesting a marginal effect.

Based on the regression model, the predictive equation can be expressed as:

$$\text{Final_Grade} = -2.918 + 0.407 \cdot \text{Gender} + 0.465 \cdot \text{Background} + 0.586 \cdot \text{HS_Phys_Avg} + 0.031 \cdot \text{Attendance_Percentage} + 0.727 \cdot \text{MethodUsed}$$

This equation provides an indication of the direction and relative strength of associations between the examined variables and students’ final grades within the studied sample.

3.1 Qualitative Evaluation of Students’ Reflections

In addition to the quantitative analysis, a qualitative evaluation was conducted based on reflections collected from 41 students through open discussions and written comments at the end of the course. The analysis followed a descriptive approach, focusing on recurring patterns and representative statements to gain insight into students’ perceptions of the impact of Python and Jupyter Notebooks on their engagement and conceptual understanding in Physics.

3.1.1 Improved Conceptual Understanding and Visual Perception

Students consistently reported that the use of Python helped them bridge the gap between theoretical concepts and practical applications. One student stated:

“I understand much better when I see it graphically. I can better explain what is happening.”

These reflections suggest a shift from passive reception of information toward active exploration and cognitively engaged learning, which is consistent with constructivist perspectives in physics education literature [2, 15]. The use of visual simulations appeared

to strengthen students' ability to interpret physical phenomena and articulate underlying mechanisms.

3.1.2 Increased Engagement and Interaction

Many students emphasized that activities conducted through Jupyter Notebooks were engaging and intellectually stimulating:

“One of the few classes where I didn't feel like a passive listener, but an active participant.”

This heightened engagement was also evident in increased classroom interaction, more frequent questions, and active participation in discussions. These observations are in line with previous findings reported by Basholli et al. [10], who emphasize the role of Python-based instruction in enhancing student interaction and collaborative engagement.

3.1.3 Development of Algorithmic Thinking

The structured nature of coding activities encouraged students to approach physics problems through logical, step-by-step reasoning, thereby strengthening their algorithmic thinking skills. As one student noted:

“I learned to solve physics problems not just using formulas, but step-by-step through coding.”

This shift suggests that computational modeling supported a deeper procedural understanding of physical laws and problem-solving strategies.

3.1.4 Encouragement of Independent Work

An additional observation was that several students continued using Python independently after the completion of the course, motivated by personal interest or connections to their academic programs:

“After class, I really enjoyed modeling similar situations on my own.”

Such behavior indicates a shift from externally driven participation to intrinsic motivation, which is widely recognized as a key factor in long-term academic development and sustained engagement.

Overall, the qualitative findings complement the statistical results, demonstrating that the integration of computational tools not only improves measurable academic performance but also positively influences students' engagement, motivation, and relationship with the subject matter.

4 Conclusions

The findings of this study demonstrate that integrating the Python programming language and interactive environments such as Jupyter Notebooks and GitHub into Physics instruction has a statistically significant and pedagogically meaningful impact on student learning

outcomes. Students exposed to this approach achieved significantly higher scores in midterm examinations and final grades ($p < 0.001$), confirming the effectiveness of computationally supported instruction compared to traditional methods [5, 17].

Beyond measurable academic performance, students reported that Python-based simulations and visualizations facilitated stronger connections between theoretical concepts and practical applications, while also fostering the development of visual and algorithmic thinking skills [14, 12]. The instructional approach encouraged active engagement and independent exploration, supporting a transition from passive knowledge reception to constructive learning, as emphasized by Papert [15] and Sherin [2].

Although class attendance did not differ significantly between instructional methods ($p = 0.077$), qualitative reflections indicated increased motivation, interaction, and participation during sessions that incorporated computational tools. These findings align with previous research highlighting the transformative potential of coding in science education [9, 10]. Although the findings are limited to a single institutional context and cohort, they provide empirical support for the pedagogical value.

In conclusion, the integration of Python and related digital tools should not be regarded merely as a technological enhancement, but rather as a pedagogical strategy that aligns Physics education with the competencies required in contemporary scientific and professional contexts. By promoting knowledge construction, interdisciplinary skill development, and active learning, this approach contributes to the modernization of teaching practices and better prepares students for the demands of technology-driven environments.

We gratefully acknowledge the financial support from “Fan S. Noli” University, University of Korça, which made this research possible.

All the dataset generated during the implementation of the teaching method is available from the corresponding author upon reasonable request. All personal identifiers have been removed to ensure participant confidentiality.

The contribution of the authors to this work is as follows: conceptualization, L. K., E. M., and I. S.; methodology, L. K., E. M., and I. S.; software, L. K.; writing - original draft preparation, L. K.; writing - review and editing, L. K., E. M., and I. S.

References

1. B. L. Sherin, A. A. diSessa, D. Hammer, Dynaturtle Revisited: Learning Physics through Collaborative Design of Active Simulations. *Interact. Learn. Environ.* **3**(2), 91-118 (1993). <https://doi.org/10.1080/1049482930030201>
2. B. L. Sherin, How Students Understand Physics Equations. *Cognition and Instruction. Cogn. Inst.* **19**(4), 479–541 (2001). https://doi.org/10.1207/S1532690XC11904_3
3. S. Alabidi, K. Alarabi, H. Tairab, Sh. Alamassi, N. R. Alsalthi, The effect of computer simulations on students’ conceptual and procedural understanding of Newton’s second law of motion. *Eur. J. Math. Sci. Technol. Educ.* **19**(5), (2023). <https://doi.org/10.29333/ejmste/13140>
4. J. Weber, Th. Wilhelm, The benefit of computational modelling in physics teaching: a historical overview. *Eur. J. Phys.* **41**, 034003 (2020). <https://doi.org/10.1088/1361-6404/ab7a7f>
5. N. D. Finkelstein, W. K. Adams, C. J. Keller, P. B. Kohl, K. K. Perkins, N. S. Podolefsky, S. Reid, When learning about the real world is better done virtually: A

- study of substituting computer simulations for laboratory equipment. *Phys. Rev. Phys. Educ. Res.* **1**(1), 010103 (2005). <https://doi.org/10.1103/PhysRevSTPER.1.010103>
6. N. D. Finkelstein, K. K. Perkins, W. K. Adams, P. B. Kohl, N. S. Podolefsky, Can computer simulations replace actual equipment in undergraduate laboratories? *AIP Conf. Proc.* **790**(1), 101-104 (2005). <https://doi.org/10.1063/1.2084711>
 7. S. J. Kumar, Effects of Computer Simulations on Senior Secondary School Students' Achievements in Practical Physics in Educational District III, Lagos State, Nigeria. *Glob. J. Human-Soc. Sci. Res.* **18**(8), (2018). https://socialscienceresearch.org/index.php/GJHSS/article/view/101736/3-Effects-of-Computer-Simulations_JATS_NLM_xml#toc
 8. P. Rebusco, A. Barrantes, B. McGimsey, L. Royden, Python with First Year Physics: What we Taught and What we Learned. (MITFaculty Newsletter, 2017). https://web.mit.edu/fnl/volume/302/rebusco_et%20al.html
 9. A. Krakowski, E. Greenwald, N. Roman, Ch. Morales, S. Loper, Computational Thinking for Science: Positioning coding as a tool for doing science. *J. Res. Sci. Teach* (2023). <https://doi.org/10.1002/tea.21907>
 10. F. Basholli, D. A. Juraev, Rr. Ormeni, Edukimi Inhinierik per Inteligjencen Artificjale Dhe Mesimin E Makinerive Nepermjet Programimit ne Python. *Optime*, (2) 507-515 (2024). <https://doi.org/10.55312/op.vi2.6188>
 11. L. A. Barba, L. J. Baker, D. S. Blank, j. Brown, A. B. Downey, T. George, L. J. Heagy, K. T. Mandli, J. K. Moore, D. Lippert, K. E. Niemeyer, R. R. Watkins, R. H. West, E. Wickes, C. Willing, M. Zingale., Teaching and Learning with Jupyter, Open Book, (2019). <https://jupyter4edu.github.io/jupyter-edu-book/>
 12. F. Perez, B. E. Granger, IPython: A System for Interactive Scientific Computing. *Comput. Sci. Eng.* **9**(3), (2007). <https://doi.org/10.1109/MCSE.2007.53>
 13. A. Mandanici, G. Mandaglio, G. Pirrotta, V. C. Nibali, G. Fiumara, Simple Physics With Python: A Workbook on Introductory Physics With Open-Source Software. *Comput. Sci. Eng.* **24**(2), (2022) <https://iris.unime.it/retrieve/7be74bfa-9606-4cb5-9aff-e99a475bdd65/3232251.pdf>
 14. T. De Jong, Technological Advances in Inquiry Learning. *Science. Science.* **312**(5773), (2006). <https://doi.org/10.1126/science.1127750>
 15. S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. ISBN: 978-0-465-04627-0 (Basic Books, Inc. Division of HarperCollins, New York, 1980) p. 5.
 16. D. Sands First year mechanics taught through modelling in VPython. *New Dir. Teach. Nat. Sci.* **49**(6), (2010). <https://doi.org/10.29311/ndtpps.v0i6.383>
 17. M. F. H. Moyano, N. B. Garcia, M. C. B. Saavedra, M. A. S. Paguay, A. F. E. Luzuriaga, J. M. B. Borja, S. F. H. Moyano, Using Python for Teaching Classic Mechanics in University Students. *Russ. Law J.* **11**(8), (2023). <https://russianlawjournal.org/index.php/journal/article/view/1299/724>