Caliphate Journal of Science & Technology (CaJoST)

ISSN: 2705-313X (**PRINT**); 2705-3121 (**ONLINE**) **Research Article**

Open Access Journal available at: https://cajost.com.ng/index.php/files and https://www.ajol.info/index.php/cajost/index

DOI: https://dx.doi.org/10.4314/cajost.v6i1.5

Article Info

Received: 17th September 2023

Revised: 10th January 2024

Accepted: 13th January 2024

Comparative Evaluation of Python Integrated Development Environments (IDEs) on Android Mobile Devices: Implications for Teaching Programming Concepts

Adaiti A. Kadams

Department of Computer Science, Modibbo Adama University, Yola, Nigeria.

***Corresponding author's email:**

adaiti@mau.edu.ng.

Programming is a major part of software engineering. There are many programming languages, paradigms, frameworks and Integrated development environment (IDE) software's used to teach programming concepts today. Access to personal computers or computer laboratories is poor or non-existent for students in developing countries due to socio-economic challenges. One useful resource is the growing availability of cheaper Android mobile devices in developing countries. With the right applications and editors, these mobile devices can be used to practically teach programming with hands-on programming experience by students of these developing countries. This article provides a comparative analysis of Python Integrated Development Environments (IDEs) available for mobile Android devices, focusing on their suitability for teaching programming concepts. The evaluation is conducted on five popular Python IDEs tailored for Android mobile devices. It provides a comprehensive comparison of five popular mobile Python programming IDE applications. The comparison covers their respective advantages and disadvantages, ease of use, installation, features, performance, Central Processing Unit (CPU) and Random-Access Memory (RAM) resources usage. The study highlights the suitability of Pydroid 3 and QPython 3x/Ox for novice learners operating resource-limited Android devices, whereas AIDE and Termux cater to individuals with intermediate to advanced proficiency, particularly those equipped with more powerful Android devices. This discernment serves as a valuable resource for educators and learners alike, aiding them in making informed decisions when choosing the optimal tool for imparting and acquiring programming concepts through Python on Android mobile platforms.

Keywords: IDEs, interpreter, Compiler, Python, Android, Pedagogical-technologies.

Cite this: *CaJoST*, 2024, 1, 35-44

1. Introduction

Software development is increasing becoming popular amongst young people interested in creating software applications. Software is a part of everyday life, from entertainment to professional productivity [1]. Educators all over the world are using several resources on the internet to teach and tutor students on how to write programs. The traditional way of teaching programming in a class is still a valid Pedagogy model when there are available resources for the students to practice and follow. Regrettably, this is not the situation in numerous schools within developing countries

As discussed by Oroma et al. [2], teaching programming to novice students in developing countries presents challenges such as pedagogical approaches, insufficient infrastructure, limited access to up-to-date or state-of-the-art resources, untrained educators, socioeconomic disparities, unreliable electrical supply, variations in students' educational backgrounds, and financial limitations. Dasuki and Quaye [3] observed that these challenges contribute to subpar performance and high failure rates

among students in computing courses schooling in developing countries.

However, a potential solution for the shortage of technological resources, including laptops, desktops, electricity, and workstations, can be found in the widespread availability of mobile devices, particularly Android smartphones, in developing nations, notably across Africa. The expulsion of cheaper internet access and android mobile devices from Asia has made access to these devices possible for so many students in schools within these developing countries as studies by Poushter et al [4].

Mobile phones with internet access are prevalent among students in universities and secondary schools in Nigeria. As internet access becomes increasingly affordable and accessible to these students, there is potential to utilize these mobile devices as an alternative technology for teaching and learning programming, even in schools located in remote areas.

2. Related Works

An essential aspect of teaching and learning software development among young people is emphasizing on hands-on experiences in building real-world applications, interactive learning, and a learner-centric approach. Machalík and Jahodová Berková in 2017 [5] found that the conventional approach to teaching programming languages in developed nations is often costly and inaccessible to students in many developing countries. However, many of these students have access to modern mobile devices, fuelling their interest in creating software applications.

Waite and Sentence in 2021 [6] demonstrated that regardless of the instructional method employed by the educator, hands-on practice and active participation are the most effective means for beginners to grasp programming concepts. The use of Pedagogical technologies by an educator to apply their logical knowledge and the ability to apply it practically on the computer for students to understand cannot be understated. Practical teaching approaches like the Extreme Apprenticeship proposed by Vihavainen et al. [7], emphasize learning by doing together with continuous feedback as the most efficient means for learning. This shows that researchers and lecturers have noted the importance of situating students by linking what they have been thought with practical executions of real-world examples. Mobile devices are considered by Chung et al. [8] as a practical way of allowing students to acquire hands-on experience, rather than being a medium to only pass over information.

Prior research has also explored the use of mobile devices to teach different programming languages in an educational setting, such as the concept of teaching programming on mobile devices in the future as proposed by Tillmann et al. [9]. Sakibayev et al. [10] introduced the use of Android-based mobile device as a programming environment, though there's limited focus on Python IDEs for low-end Android smartphones in developing countries.

Furthermore, while studies like Werner [11] emphasize the 'how' of mobile learning in teaching programming, De Silva KK et al. [12] proposed the use of Distributed Integrated Development Environment for Mobile Platforms (DIMP) to enable collaborative programming on mobile phones or tablet computers. This technology could be beneficial in teaching and learning environments where students have access to Android phones with steady internet resources as this technology consist of a central server and a set of compilation servers in the cloud.

Alanazi et al. [13] demonstrated through their research that mobile devices equipped with programming Integrated Development Environment (IDE) applications positively influence the learning experiences of Saudi female students in programming subjects. The study underscored the significant effectiveness of integrating mobile devices to enhance learners' comprehension and engagement in programming.

While various studies explore the benefits and potential of utilizing mobile devices for teaching programming, there remains a noticeable gap in the literature regarding comparative research on IDE applications. Specifically, there is limited research focusing on the needs of learners and educators operating in environments with constrained resources for education.

3. Methodology

In this study, the comparison covers their respective advantages and disadvantages, ease of use, installation, features, performance, Central Processing Unit (CPU) and Random-Access Memory (RAM) resources usage, educational suitability and community or developer support. Each tool is also scrutinized for its unique approach and its potential to enhance the instruction of programming concepts. We considered IDEs such as Termux, Pydroid 3, QPydriod 3L, Pyonic Python 3 Interpreter and AIDE for Android OS Mobile devices. There are other mobile IDEs in beta versions that are not accessible and have a lot of bugs out there. These five IDEs are popular, essentially free and readily available for educational purposes. However, some are not totally free for advanced programming. An IDE like PyCharm mobile is another example of a robust Android Integrated Development Environment (IDE). It comes in two desktop editions: Community and Professional. The Community edition, being an open-source project, is free but offers fewer features. On the other hand, the Professional edition, available commercially, boasts an exceptional array of tools and features. Furthermore, the PyCharm IDE currently in the android Google Play Store doesn't function and is not by developed by Jet-Brains.

'Memu Play' was an emulator used to run all the mobile Integrated Development Environments (IDEs) on a computer system. 'Simple Systems Monitor' application was used to test each mobile Integrated Development Environments (IDEs) based on its usage of mobile device RAM and CPU resources. Ahdan et al. [14] demonstrated that mobile device RAM and CPU resources are basically the most essential parts of the performance of any mobile device which will be accessible to students in developing countries.

The Android mobile Integrated Development Environments (IDEs) applications were tested by running the same types of python programs. For the purpose of this study the factorial snippet was used to run a large integer as 9,999. This gave the Simple Systems Monitor application more time to study the mobile device's RAM and CPU usage by the various android mobile Integrated Development Environments (IDEs).

3.1 Termux

Termux, as defined on the developer's website [15] is an Android terminal emulator that works with no rooting or setup needed. Termux offers a dark interface which is not changeable and yet an added advantage in saving mobile device battery energy and life. This also adds less strain to eye-sight due to the low light emission from the unchangeable interface as observed by Eisfeld and Kristallovich [16]. Its interface layout is similar to a Linux terminal window, giving it a techy feel. An example of Termux application running in Figure 1. shows what it looks like on a mobile device.

The Termux IDE is a straight forward application to use with the following advantages [15]:

- ✓ Secured access to remote servers is ensured through the utilization of the OpenSSH client.
- ✓ API endpoints are readily accessible through the use of the curl command.
- ✓ Installation via the APT package management system presents a slightly higher technical complexity.
- ✓ It boasts a portable Python console enhanced by read-line capabilities.
- ✓ Termux provides robust support for keyboard shortcuts and accommodates mouse or stylus input.
- ✓ It extends its compatibility to other programming languages, including Ruby, Perl, and Node.js.

Termux serves as a valuable IDE for instructing fundamental and advanced Python programming concepts, suitable for both novice learners and university students, once the initial installation challenge is overcome.

Figure 1. The Termux application interface for a mobile device emulator

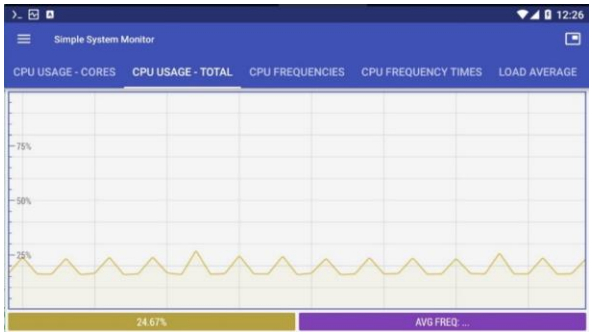

Figure 2. Termux application: Total CPU Usage of 24.67% on average while running.

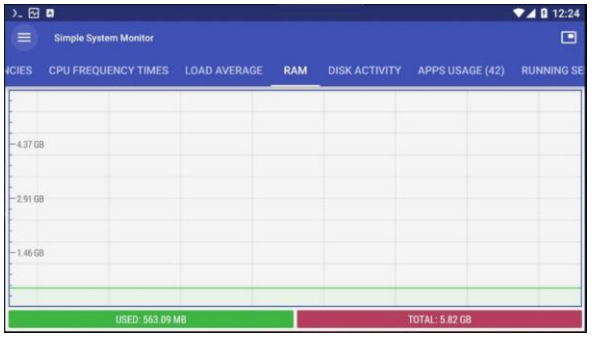

Figure 3. Termux application: Total RAM Usage of 563.09 megabytes from 5.82 gigabytes of RAM, making 9.68% of total RAM usage while running the same python snippet.

3.2 Pydroid 3

Pydroid 3 is an IDE for python 3. Pydroid 3 provides an integrated development environment (IDE) for Python programming. Students can write, run and edit python code on Andriod mobile devices with low CPU and RAM resources [17]. It has features such as syntax highlighting, code prediction and debugging tools to enable smooth python coding on any android mobile device.

Pydroid 3 IDE has the following key features and properties;

- ✓ Installation is simple and requires minimal additional steps.
- ✓ It enables the use of tkinter Graphical User Interface (GUI) for creating executable user interfaces on android mobile devices.
- ✓ Additional modules or libraries can be easily integrated using 'pip install.'
- ✓ Features auto indentation and code analysis, similar to an integrated development environment (IDE) on a desktop or laptop computer.
- ✓ It mandates 200MB of free internal memory, with an additional 20MB for bulkier libraries when added.
- ✓ Delivers outstanding debugging capabilities, allowing students to establish breakpoints, step through code, and examine variables.

Overall, Pydroid 3 is a comprehensive tool for students or anyone who wants to write, test, and debug python code on their android mobile devices. The file management capacity is straight forward. Notably, it possesses a streamlined set of features that supports efficient and effective python programming in any android mobile environment with smartphones in mind.

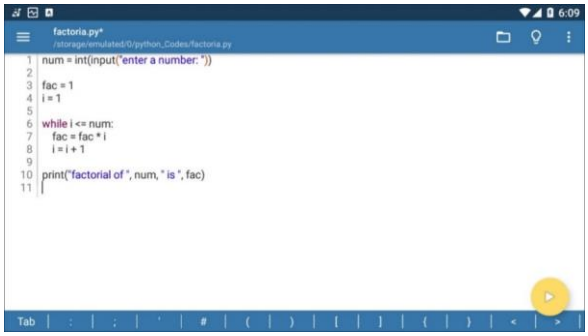

Figure 4. Pydroid 3 IDE: Appealing mobile interface, with a bottom tab for syntax.

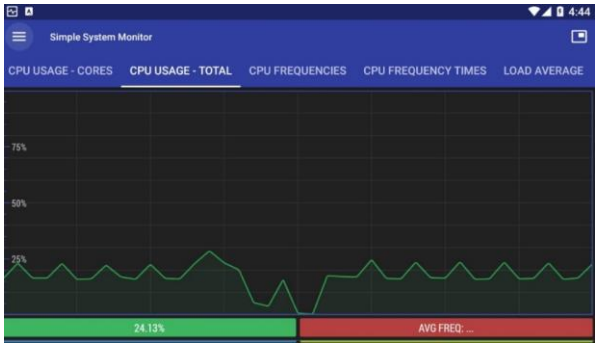

Figure 5. Pydroid 3 IDE: The Total CPU Usage of 24.13% on average while running the test factorial python snippet.

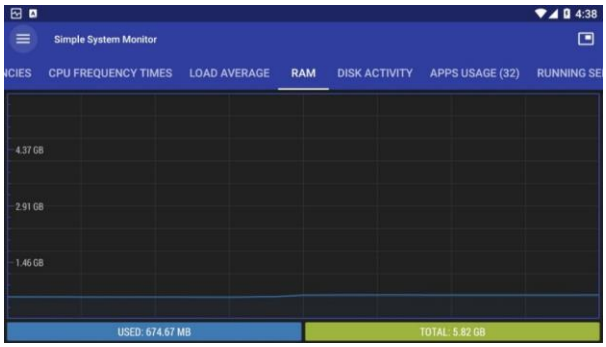

Figure 6. Termux application Total RAM Usage of 674.67 megabytes of 5.82 gigabytes of RAM (11.59% of total available RAM) on running the same python snippet.

3.3 QPython 3L

Also called QPython. QPython 3L is a fairly new player in the android mobile device python IDE arena and yet has millions of users worldwide [18]. QPython is the Python engine for android and still in Beta version. It contains features such as, python interpreter, runtime environment, editor and SL4A library.

There are two version of the QPython 3L IDE; the QPython Ox and QPython 3x. The QPython Ox is designed with younger beginners in mind, as it offers a multitude of user-friendly features that facilitate effortless learning and comprehension. The QPython 3x is for much more matured and advanced learners.

QPython 3L has some viable features such as;

- ✓ Streamlined interface for beginners to easily access and learn python code.
- ✓ Supports running different projects such as console programs, SL4A programs, game development programs and webapp programs.
- ✓ Utilises the QR code reader to exchange codes with other Android mobile devices.
- ✓ The capacity to support scientific libraries, such as numpy, scipy, matplotlib and scikit-learn.
- ✓ Easy to use editor for writing, editing and displaying python code.
- ✓ QPython 3L has a very good customer support and documentation as it is still in its beta version.

Positive ratings on the Google Play Store indicate that QPython 3L is a strong competitor among popular mobile Python IDEs. However, ongoing development and a number of existing bugs are still noteworthy aspects of this Python mobile IDE. Figure 7 illustrates the interface of the QPython 3L mobile application.

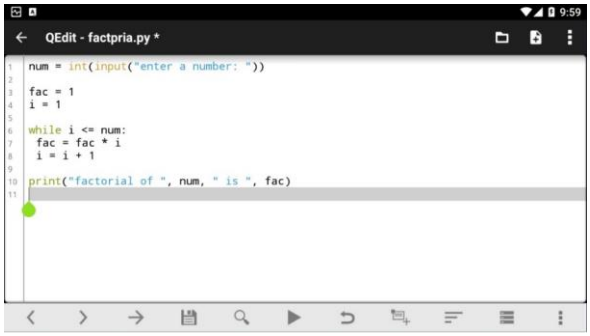

Figure 7. QPydroid 3L: Android mobile device interface with python snippet.

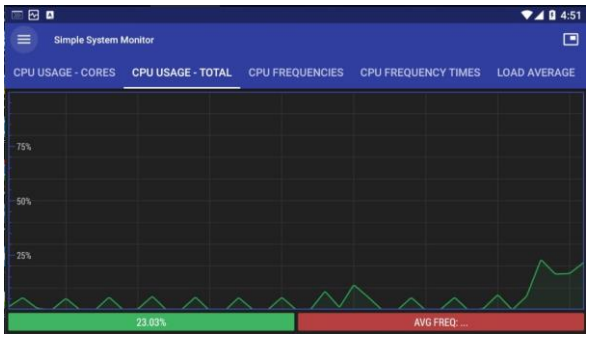

Figure 8. QPython 3L: Total CPU Usage of 23.03% on average while running python snippet.

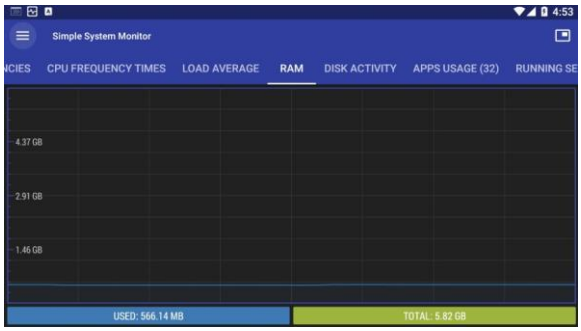

Figure 9. QPython 3L: Total RAM Usage of 566.14 megabytes of 5.82 gigabytes of RAM which is 9.73% of total available RAM on running the same python snippet.

3.4 Pyonic Python 3 Interpreter

The Pyonic Python 3 Interpreter is an IDE because it has a functional graphical user interface (GUI) for the Android mobile Python Interpreter to code, test, and debug. Interpreter integrated IDEs are lighter on mobile CPU and RAM usage when running programming snippets compared to Python IDEs integrated with compiler capabilities, which have a broader range of capabilities [19]. The Pyonic Python 3 Interpreter has earned a strong reputation, and it offers the following key features:

- ✓ Offers compatibility with both Python 2 and Python 3 versions.
- ✓ The Pyonic 3 Interpreter IDE's graphical user interface (GUI) is developed using Python Kivy, allowing inputs to be routed to a secondary interpreter through Android services, executed as a separate process.
- ✓ Introduces customizable button rows (numbers, additional symbols, etc).
- ✓ Provides support for a variety of file types.
- ✓ Offers nearly precise code completion.
- ✓ Encounters issues on certain Android devices, including selected models from Nexus, Sony Xperia, and Samsung, where functionality may be compromised.

The method used by Pyonic Python 3 Interpreter to run code is a bit crude. There is the issue of poor support from many high-end android mobile devices. These products are not financially affordable for most students in developing countries. The compatibility issue can be ignored. Figure 10. shows the Pyonic Python 3 Interpreter android mobile device interface. It's not different in display with other powerful python IDEs. Nevertheless, it has support for many python libraries that are limited and difficult to integrate. It also has a drawback of only supporting older versions of the Android operating system.

Figure 10. Pyonic Python 3 Interpreter android mobile device interface.

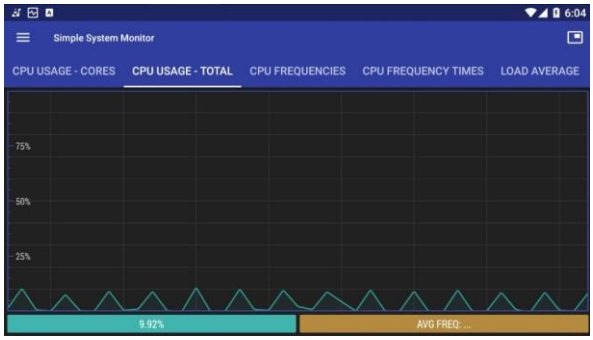

Figure 11. Pyonic Python 3 Interpreter: Total CPU Usage of 9.92% on average while running python snippet used for evaluation.

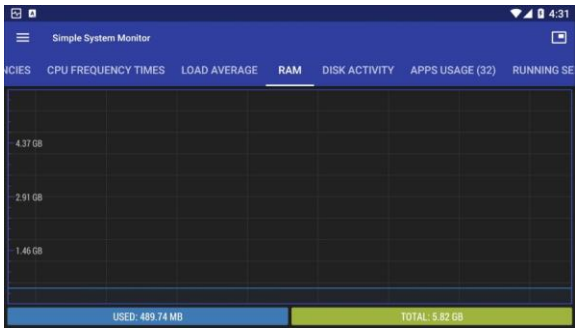

Figure 12. Pyonic Python 3 Interpreter: Total RAM Usage of 489.74 megabytes of 5.82 gigabytes of RAM (8.41% of total available RAM) on running the same python snippet used for evaluation of the CPU usage capacity.

3.5 AIDE (Android IDE)

AIDE refers to Android IDE (Android Integrated Development Environment), which

is an integrated development environment for Android app development. It is a powerful tool for programming in many languages, such as Java and C/C++. Bernard [20] has demonstrated that AIDE has a diverse set of tools to accommodate the entire development lifecycle of an application, i.e., from writing code, debugging to testing and deployment.

Some of the major features of AIDE are;

- ✓ Code Editing and Autocompletion: The syntax highlighting and code completion greatly assist students in writing clean code.
- ✓ Debugger: A debugger helps in finding and fixing bugs.
- ✓ Version Control Integration: It supports systems like Git, allowing students to manage and track changes effectively.
- ✓ AIDE has Performance Analysis tools such as; Battery Usage Analysis, CPU Profiling, Memory Profiling and App Launch Time Analysis.
- ✓ Sadly, AIDE is not easily available for python. However, with some technicality it can be integrated.

While AIDE offers great versatility, it does demand significant resource utilization. It is most suitable for students transitioning into professional programming roles, particularly those who seek to create functional Android apps while on the move.

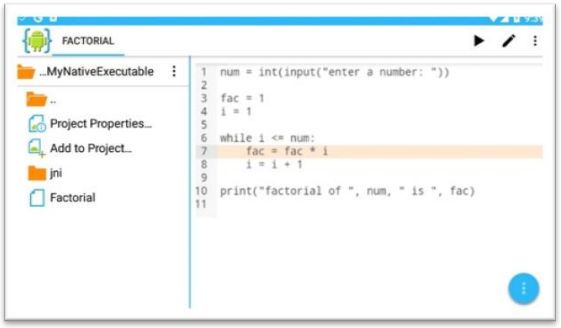

Figure 13. AIDE (Android IDE) mobile device interface.

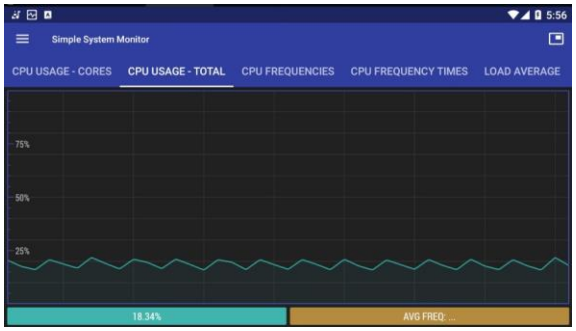

Figure 14. AIDE (Android IDE): Total CPU Usage of 18.34% on average while running python snippet used for evaluation.

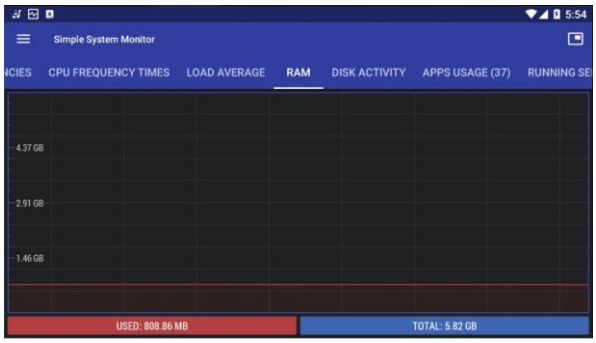

Figure 15. AIDE (Android IDE): Total RAM Usage of 808.86 megabytes of 5.82 gigabytes of RAM (13.90% of available total RAM) on running the same python snippet used for evaluation of the CPU usage capacity

4. Comparison of Python IDEs for Android Devices

Table 1 provides a comparative analysis of Python IDE applications for Android devices, focusing on their RAM and CPU usage in relation to the device's available resources. It's important to note that the percentage values may vary depending on the processor's capacity and RAM size. CPU and RAM utilization are critical factors for ensuring a smooth experience when running and learning Python programming on Android mobile devices.

	Termux	Pydroid 3	QPython 3L	Pyonic Python 3	AIDE (Android IDE)
CPU Usage	24.67%	24.13%	23.03%	9.92%	18.34%
RAM Usage	563.09MB	674.67MB	566.14MB	489.74MB	808.86MB

Table 1. Python IDEs comparison for five (5) android applications

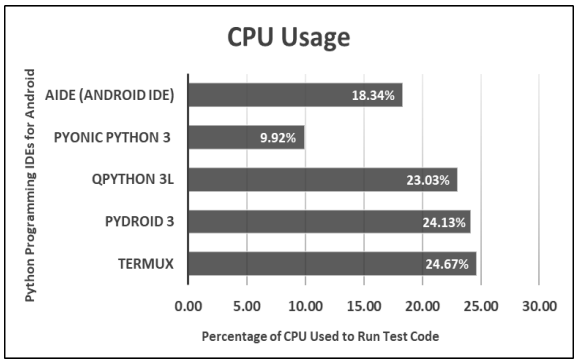

Figure 16. A graphical comparison and visualization of CPU usage across different Python programming IDEs designed for Android mobile devices.

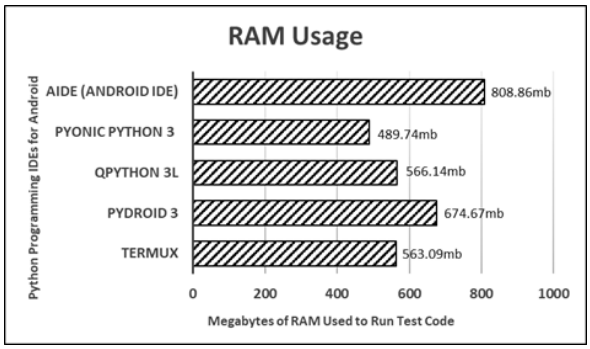

Figure 17. A graphical comparison and visualization of RAM usage across different Python programming IDEs designed for Android mobile devices.

AIDE (Android IDE) distinguishes itself by its requirement for higher RAM resource (see Figure 17) compared to most mobile IDEs, albeit offering a trade-off with lower CPU utilization (see Figure 16). While RAM capacity remains a crucial consideration, users with devices featuring 1GB of RAM may need to close concurrent applications to ensure a seamless experience with the AIDE application. An important feature is AIDE's compatibility with Java and C++ coding, positioning it as an essential tool for students embarking on a broader spectrum of programming languages.

The users of Pyonic Python 3 will experience notably efficient CPU usage, with a relatively modest RAM footprint. However, the application's drawback lies in limited bug support, as its core functions as an interpreter with a Python Kivy GUI for coding, testing, and debugging, making it a limited IDE. Extending library support may pose technical challenges and might not encompass a wide range of Python programming modules as explained in [19]. Additionally, the limited device support of Pyonic Python 3 may hinder its adoption among students seeking a versatile learning platform.

QPython 3L, while emerging as a relatively new entrant, and still in its beta version, has garnered a substantial global user base [18]. It exhibits a higher CPU utilization rate, but with a proportionately lower RAM demand. An inherent advantage is its dual version availability—QPython 3x and QPython Ox—making it adaptable to students at varying skill levels. The QPython Ox version, in particular, serves as an accessible entry point for beginners, including those transitioning from unrelated fields of study.

Pydroid 3 emerges as a robust Python programming IDE, empowering students to write, run, and edit Python code. This functionality, however, comes at the expense of relatively elevated CPU and RAM consumption, as depicted in Figure 16. Nevertheless, Pydroid 3 offers significant benefits, such as auto indentation, robust debugging capabilities, and proficient code prediction. The inclusion of pygame module and tkinter GUI support further encourages students to build interactive python programmed applications and games with event-driven functionality.

Contrastingly, Termux exhibits the highest CPU utilization, offset by a moderate RAM requirement, as highlighted in Figure 16. It's noteworthy that Termux is not accessible via the Google app store and necessitates downloading and installation from the developer's website. The technicalities associated with its installation set it apart from other Android IDE applications. An attractive feature is its compatibility with a range of programming languages, including Ruby, Perl, and Node.js."

This revised passage provides a more structured and academically oriented presentation of the information.

5. Conclusion

The domain of Python Mobile Integrated Development Environments (IDEs) offers a diverse array of options tailored to various user needs, especially those who require mobility. Understanding the nuances and implications of these tools is crucial, particularly in the context of students in developing countries grappling with intermittent electricity supply.

For students in resource-constrained settings, Pydroid 3 and QPython 3x/Ox emerge as commendable options at the foundational level. These applications feature straightforward user interfaces and moderate CPU and RAM requirements. Additionally, extensive developer support enhances their suitability for novice users.

Conversely, AIDE and Termux gravitate toward the intermittent or advanced learners. Their robust functionalities and capabilities are better harnessed by those well-versed in the intricacies of Python programming, especially in environments with stable power sources.

Pyonic Python 3 Interpreter assumes a distinctive role by accommodating only a limited selection of Python libraries, which can be arduous to install. It serves as a viable platform for composing concise Python snippets and facilitating print style debugging, albeit its scope is relatively narrower.

In conclusion, it is pertinent to underscore the potential for further comparative research. Exploring the collaborative capabilities of these IDEs among students and a closer assessment of the non-functional requirements, such as security, usability, and performance. This would contribute to a more comprehensive understanding of the overall suitability of these IDEs in diverse educational contexts.

While this study primarily focused on the general use, features, CPU and RAM resource usage, and educational suitability trends of five popular Android Python IDEs for mobile devices, there is room for further exploration. Future research endeavors could delve into the collaborative capabilities of these Python IDEs, particularly in fostering collaboration, version control, sharing, and synchronization among students and educators. This qualitative investigation aims to gather insights directly from educators and students, shedding light on the non-functional requirements of each application. By providing firsthand experiences, this additional research would offer a holistic perspective for informed decision-making when selecting a Python IDE for Android mobile devices in teaching, learning, and collaborative contexts.

Conflict of interest

The author declares no conflict of interest.

Acknowledgements

The author expresses gratitude to the reviewers for their valuable and constructive input during the paper's revision process, as it has significantly enhanced the quality of the work.

References

[1]. Fronza I, El Ioini N, Corral L. Students want to create apps: leveraging computational thinking to teach mobile software development. In Proceedings of the 16th annual conference on information technology education 2015 Sep 29 (pp. 21-26).

[2]. Oroma J, Wanga H, Ngumbuke F. Challenges of teaching and learning computer programming in a developing country: lessons from Tanzania. In INTED2012 Proceedings 2012 (pp. 3820-3826). IATED.

[3]. Dasuki S, Quaye A. Undergraduate students' failure in programming courses in institutions of higher education in developing countries: A Nigerian perspective. The Electronic Journal of Information Systems in Developing Countries. 2016 Sep;76(1):1-8.

[4]. Poushter J, Bishop C, Chwe H. Social media use continues to rise in developing countries but plateaus across developed ones. Pew research center. 2018 Jun 19;22:2-19.

[5]. Machalík S, Jahodová Berková A. Modern trends in teaching programming on mobile devices. In: L. Gómez Chova, A. López Martínez, and I. Candel Torres, editors. ICERI2017 Proceedings. Seville, Spain: IATED; 2017. p. 8904-8909.

[6]. Waite J, Sentance S. Teaching programming in schools: A review of approaches and strategies. Raspberry Pi

Foundation. 2021 Nov.
[7]. Vihavainen A, Paksula M, Luukkainen M. Extreme apprenticeship method in teaching programming for beginners. In Proceedings of the 42nd ACM technical symposium on Computer science education 2011 Mar 9 (pp. 93-98).
[8]. Chung CJ, Hwang GJ, Lai CL. A review of experimental mobile learning research in 2010–2016 based on the activity theory framework. Computers & education. 2019 Feb 1;129:1-3.
[9]. Tillmann N, Moskal M, De Halleux J, Fahndrich M, Bishop J, Samuel A, Xie T. The future of teaching programming is on mobile devices. InProceedings of the 17th ACM annual conference on Innovation and technology in computer science education 2012 Jul 3 (pp. 156-161).
[10]. Sakibayev S, Sakibayeva B, Toibazarov D. Android-based mobile device as a programming environment in a school programming class. Indian Journal of Science and Technology. 2016;9(20):94488-.
[11]. Werner M. Teaching graphics programming on mobile devices. Journal of Computing Sciences in Colleges. 2013 Jun 1;28(6):125-31.
[12]. De Silva KK, Darshana PHRA, Silva WADM, Basnayake BMNA, Wickramarathne CJ. Distributed Integrated Development Environment for Mobile Platforms. In: 8th International Conference on Computer Science & Education; 2013; Colombo, Sri Lanka. IEEE; 2013. p. 324-329. doi:10.1109/ICCSE.2013.6553932.
[13]. Alanazi A, Li A, Soh B. Effects on Saudi Female Student Learning Experiences in a Programming Subject Using Mobile Devices: An Empirical Study. Int J Interact Mobile Technol. 2023;17(13).
[14]. Ahdan S, Susanto E, Syambas N. Proposed Design and Modeling of Smart Energy Dashboard System by Implementing IoT (Internet of Things) Based on Mobile Devices. In2019 IEEE 13th International Conference on Telecommunication Systems, Services, and Applications (TSSA) 2019 Oct 3 (pp. 194-199). IEEE.
[15]. Termux. Termux. https://termux.dev/en/ [Accessed October 21, 2023].
[16]. Eisfeld H, Kristallovich F. The Rise of Dark Mode: A qualitative study of an emerging user interface design trend [Internet] [Dissertation]. 2020. Available from: https://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-50563.
[17]. Pydroid 3: Educational IDE for Python 3. https://download.cnet.com/pydroid-3-educational-ide-for-python-3/3000-20414_4-78094802.html [Accessed October 21, 2023].
[18]. He F, Hu X, Liu S, Li T, Zhu K, Bao X, Jiang C. Algorithm for Improving Processor Utilization in Multi-core Processor Environment by Python Language. In2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC) 2021 Jun 18 (Vol. 4, pp. 775-779). IEEE.
[19]. Inclement. Pyonic-interpreter. https://github.com/inclement/Pyonic-interpreter [Accessed October 21, 2023].
[20]. Bernard J. AIDE--developing for Android on Android. Linux Journal. 2013 Jun 1;2013(230):1.

