

Development of Basic Python Programming Error Guidance Chatbot Prototype

Duenpen Kochakornjarupong¹, Safeera Tohmong²

Department of Computer and Information Technology
Faculty of Science, Thaksin University,
Phatthalung, Thailand
duenpen@tsu.ac.th¹

Abstract— A prototype Chat Bot has been developed with the aim of guiding learners on Python programming techniques, educating them about different types of programming errors, and teaching basic commands of the Python language. The Chat Bot is designed to automatically answer questions and provide guidance to the learners. The study aims to evaluate the usability and satisfaction of using the Chat Bot prototype developed with the Dialogflow tool for collecting rule-based data. The Chat Bot prototyping process includes planning the conversation flow using diagrams and designing bot beauty screen buttons by JSON responses line with core functionality discussions, inquiries, errors, and automated message replies with error hints. The usability and satisfaction of the Chat Bot prototype were evaluated using a sample of 34 users. The results indicate that the Chat Bot prototype was effective in providing the desired information recommendations, as the mean satisfaction score was 4.27 with a standard deviation of 0.07. Therefore, the study concludes that the developed Chat Bot prototype can effectively support learners in learning Python programming.

Keywords: *Chat Bot; Python programming; advising*

I. INTRODUCTION

Nowadays, there are many technologies available for online communication, and Chat Bot technology is one of the most convenient educational tools. Chat Bots or automated answering systems can act as assistants for learners, providing guidance and advice at all times. Programming can be challenging for learners, and classroom instruction may not be enough to improve their skills. Teachers may not be able to answer all of their questions, and some students may fall behind their peers due to a lack of assertiveness or understanding.

There is research [1] shown that compiles information on the use of chatbots in computer science learning. This research was compiled from several articles on the use of chatbots in computer science learning and analyze whether the use of chatbots can enhance learning efficiency. The results showed that the use of chatbots in computer science learning can enhance learning efficiency, as chatbots can accurately simulate conversations in short periods of time and are useful for learning. Especially for learning about

programming. Using chatbots can also help students communicate effectively with questions about learning content.

Python programming language is particularly popular in the field of data science, and many educational institutions use this language to train students in programming. However, the program descriptions are typically presented in English, which can be difficult for learners who are not proficient in the language. Additionally, when learners encounter errors in their programming, they may struggle to understand the explanations even with the help of translation programs. Teachers or teaching assistants may not have the time or resources to provide individualized support to each student.

To address these challenges, we propose the development of a Chat Bot prototype that can provide advice on basic Python programming errors. While various Chat Bots [2-5] are available to provide advice to users, they may not offer one-on-one guidance when it comes to programming errors, as shown in Table I. Therefore, the researcher believes that a Chat Bot prototype that focuses on basic programming errors in Python can be helpful for learners to improve their programming skills.

TABLE I. COMPARISON TABLE OF CHAT BOT CAPABILITIES RELATED TO PROVIDE ERRORS ADVICE ON THE BASICS OF PYTHON PROGRAMMING

ChatBot Feature	[2]	[3]	[4]	[5]	Chat Bot Prototype Providing Errors Advice on the Basics Python Programming
Communicate with many people	✓	✓	✓	✓	✓
Provide advice quickly	✓	✓	✓	✓	✓
Reply to users quickly	✓	✓	✓	✓	✓
Connect to web browser	✓	✓	✓	✓	✓
Use the system through the LINE application	✗	✗	✗	✓	✓
Free to use	✗	✓	✗	✗	✓
Describing programming errors	✓	✓	✓	✓	✓
Easy to use Richmenu to be Main Menu	✗	✓	✓	✓	✓

II. TYPE OF CHAT BOTS

A Chat Bot is a computer program designed to simulate human conversations and interact with humans in real-time through voice or text. There are two types of Chat Bots: Rule-Based Bots and AI Bots. Rule-Based Bots operate based on predetermined rules and keywords, and if a question does not match the predefined rules, the Bot may give incorrect answers or malfunction. On the other hand, AI Bots use artificial intelligence and machine learning to improve their ability to understand human language and become more intelligent over time. AI Bots have become increasingly popular, with major companies such as IBM, Microsoft, Google, Facebook, and Amazon all developing their own versions.

Chat Bots have a wide range of applications, including providing information services such as answering questions through various platforms, chat programs, websites, and applications like Alexa or Siri. They are also becoming increasingly popular in the business world, as they provide a more convenient and efficient means of customer service through live chat and online sales support. As Chat Bot technology continues to improve, its applications and capabilities will only continue to grow.

A Rule-Based Chatbot, also known as a Rule-Based Approach, follows predetermined conditions to provide responses to users. It operates on a conditional basis and can only respond to questions that have been pre-programmed into the system. Its primary goal is to fulfill as many conditions as possible to achieve its purpose. There are five main types of Rule-Based Chatbots:

1. Scripted or Quick Reply Chatbots offer users a set of predefined options to choose from. Once the user selects an option, the Chatbot provides a response in a specified format.

2. Keyword Recognition Based Chatbots allow users to type in their questions or statements. The Chatbot scans for specific keywords in the message and responds accordingly in a specified format.

3. Voice-enabled Chatbots allow users to speak their questions or commands, and the Chatbot processes the responses using voice recognition technology.

4. Hybrid Chatbots combine Scripted and Keyword Recognition Based Chatbot types. Users can either type in their questions or choose from a list of predefined options to receive relevant answers.

5. Contextual Chatbots are advanced learning Chatbots that use Machine Learning (ML) and Artificial Intelligence (AI) to memorize conversations and learn from questions encountered to improve their efficiency in answering subsequent questions.

In this paper, the Chat Bot prototype developed is a hybrid type, which combines elements of both the Scripted and Keyword Recognition Based Chatbot types. Additionally, it has some features that resemble a Contextual Chatbot, as it is equipped with Machine Learning and Artificial Intelligence capabilities to

improve its efficiency in answering questions and provide more personalized responses to users.

III. ERROR TYPES IN PYTHON PROGRAMMING

Python program errors occur when the instructions in the program are not processed correctly. When a program encounters an error, it may stop immediately and be unable to continue. There are different types of errors in Python programming. The first type is Syntax Errors, which are grammatical errors that occur when parentheses are forgotten, the wrong statement is used, or the wrong library is imported. Another type is Semantic Errors, which are logical errors that cause a program to fail or malfunction, resulting in incorrect or overused resources. Syntax errors are the most basic type of error, and they are almost always fatal, as the Python parser cannot understand the code. Logical errors occur when a programmer's problem analysis is wrong, and the program is still considered complete. Finally, Exceptions occur when the Python interpreter knows what to do with a piece of code but is unable to perform the action, such as accessing the internet without an internet connection.

While Python has numerous types of errors, only nine examples are typically given and applied at a general programming level [6].

TABLE II. 9 DIFFERENT TYPES OF ERRORS

Type of Error	Description
Exception	Common mistakes
ValueError	When assigning unprocessable values, e.g. <code>x = int('0')</code>
TypeError	When selecting a data type that cannot be used by an operator or cannot be a function of a parameter, e.g. <code>x = 'x' + 10</code>
ZeroDivisionError	when dividing by 0, e.g. <code>x = 1/0</code>
NameError	Refers to a variable name that has not yet been created.
IndexError	Set a non-real sequence number, for example <code>x = [1, 2, 3] print(x[3])</code>
KeyError	Specifies a non-existent key, e.g. <code>x = {1:1, 2:2} print(x[3])</code>
ModuleNotFoundError	When the specified module name does not exist
AssertionError	When the condition of the assert statement is false

IV. RESEARCH METHODOLOGY

I. PURPOSE OF THE STUDY

The objectives of this research is to explore and gather information on fundamental errors in Python programming, as well as to analyze these errors. Afterwards, these errors will be used to design and develop chatbots, with the aim of introducing learners to basic Python programming errors.

II. ARCHITECTURE OF CHAT BOT PROTOTYPE PROVIDING ERRORS ADVICE ON THE BASIC PYTHON PROGRAMMING

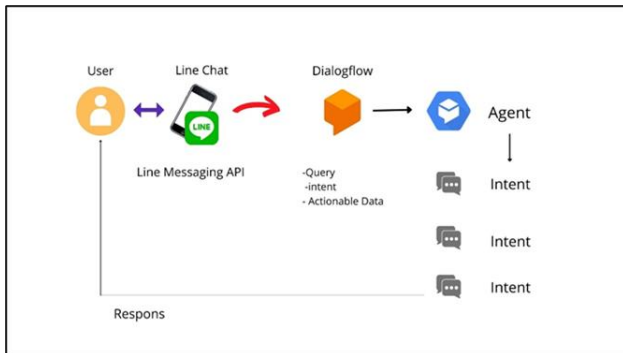


Figure 1. Architecture of Chat Bot prototype providing errors advice on the basic python programming

Figure 1 illustrates the process flow of the chatbot prototype. Users first add friends on the Line application to chat with the chatbot. The Line Messaging API is used as an admin interface with bots through Google's Dialogflow platform. Dialogflow allows for the creation of agents or bots and provides training data for machine learning and error filtering. The chatbot prototype utilizes a content-level or rule-based flowchart structure, which includes two main sections: the general user section and the admin section. In the general user section, users can type an inquiry message to receive advice on basic Python programming errors, view the main menu, or select an entry by Python programming error category type. The admin section includes the ability to type an inquiry message to provide advice on basic Python programming errors and auto-reply with information about basic Python programming errors based on the user's needs and boundary rules.

III. DESIGN AND IMPLEMENTATIONS

In designing the Chat Bot prototype, we utilized several methods including: 1) a use case diagram (Figure 2) to display the relationship between the user and all processes on the prototype, 2) use case descriptions to describe all working processes on the prototype, 3) conversation flow design to outline the overall possible conversation flows on the prototype (see Figure 3), and 4) the collection of error data during the implementation phase using Python on Google Colab. The collected error data includes the basic 9 different types of errors.

- Exception
- ValueError
- TypeError
- ZeroDivisionError
- NameError
- IndexError
- KeyError

- ModuleNotFoundError
- AssertionError

Figure 2 shows that the use case "Entry Chat Bot with QR code" is the starting point for users to open the Line application and become friends with the Chat Bot by logging in. The use case "Chat with PyHelper" refers to users chatting with the bot to ask about basic Python programming errors. If the user types a message that is not within the scope of the bot's information gathering capabilities, the bot will respond with the message "Sorry, this question is not in the scope of gathering information. Please inquire later." The use case "Use types of error menu" is for users to select the type of error they want information about or let the bot suggest errors. The category of programming errors consists of 9 types, as mentioned above.

The "Collect error data from Python Programming" use case involves the administrator collecting data types for each intent, which includes the 9 categories of error data mentioned earlier (shown in Figure 4 (a) and (b)). The administrator then collects the error information and saves it in the chatbot. The "Input data via Dialogflow" use case involves the administrator logging in via Dialogflow and entering data into each intent to add, delete, or modify data within the intent associated with programming error-gathering fields. After making changes, the administrator logs out through Dialogflow (shown in Figure 4(c) and (d)).

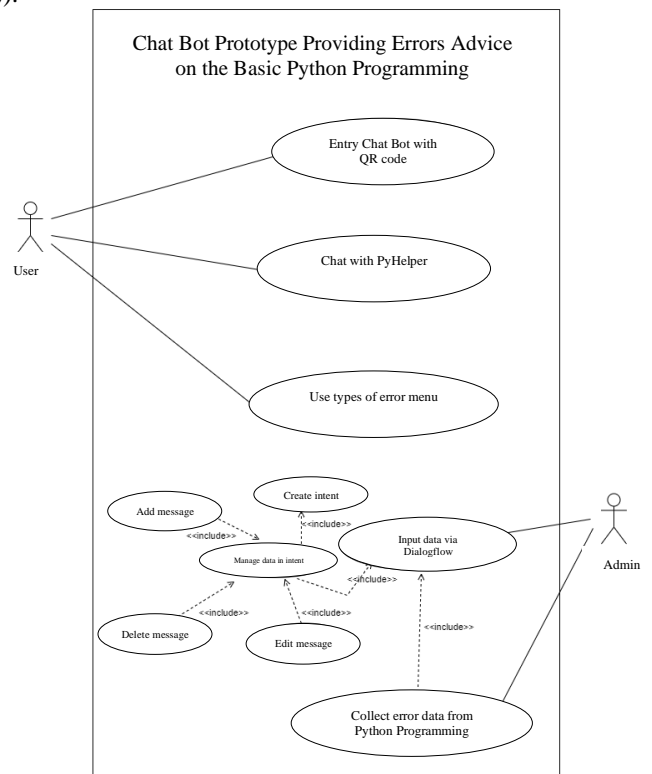


Figure 2. Use Case Diagram of Chat Bot prototype providing errors advice on the basic python programming

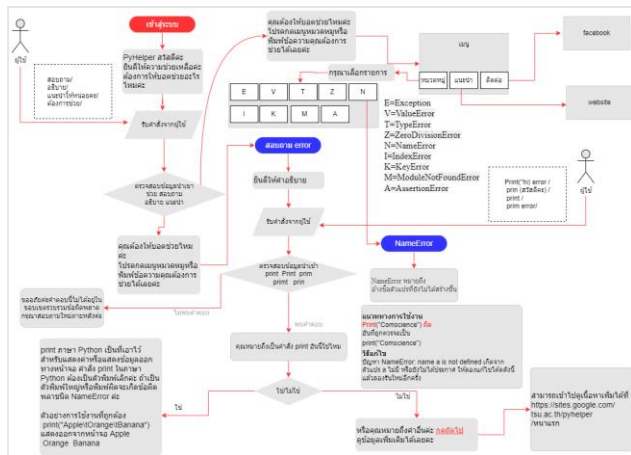


Figure 3. Overall Design of Chat Bot prototype providing errors advice on the basic python programming

IV. CHAT BOTS PROTOTYPE TESTING PLANNING

To test the effectiveness of the chatbot prototype, users were invited to interact with it and request information or recommendations. The sample population for this testing phase consisted of 5 information technology experts, 20 high school students, and 25 computer science students. The researcher used a form to evaluate the prototype, linking relevant programming error information to the users' needs.

The evaluation process involved several steps, including analyzing the flow of information, creating an analytical questionnaire, presenting the results to experts for feedback, and using the feedback to improve and refine the chatbot. Additionally, a questionnaire was created for students and users to provide their own evaluations of the chatbot prototype.

The chatbot prototype was evaluated by experts on three key aspects: convenience, design, and accuracy. Based on the feedback received, the prototype was reworked to improve its performance.

V. FINDINGS

A. Evaluating and Improving Chatbot Prototype through Expert Feedback.

The assessment of chatbot prototype by 4 experts revealed positive results in terms of convenience, design, and data accuracy. The evaluation scored a good level of 4.3 and the experts consisted of 3 computer technicians from Thaksin University and a computer teacher from Amart Panichchanukul School in Krabi Province. The chatbot was found to have easy-to-use interactive buttons, fast response time, easy-to-understand language suitable for high school students and young adults, and provided

concise information on basic Python programming techniques.

Recommendations for improving the chatbot based on expert analysis:

1. Provide feedback to users on whether their code is correct or incorrect, as the chatbot currently only provides advice when the code is incorrect.
2. Increase the amount of content information available.

Experts recommend improving the chatbot by adding basic Python commands, including:

- A command to convert data to integer data type (int).
- A command to convert data to floating point data type (float).
- Alternative statements using "if...else".
- Loop statements with a "while" condition.
- Looping commands with a "for" count.
- A reduction command for nested "elif" statements.

B. Chatbot test results with users

After improving the chatbot prototype (as examples in figure 5) to offer guidance on basic Python programming errors, we conducted real-life testing to evaluate its performance and user satisfaction. The evaluation was conducted on a specific sample of 34 individuals, including high school and computer science students, according to the testing plan. The evaluation results were analyzed using a 5-level rating scale, ranking from highest to improvement, to determine the level of user satisfaction with the chatbot prototype.

The evaluation results showed that the chatbot prototype's performance and satisfaction in real use were generally rated positively, with a score of 4.27 out of 5. The evaluation was based on 12 criteria, including convenience and access, menu format, use of interactive buttons, coverage of user needs, language level, response consistency, accuracy, convenience, response time, content conciseness, and effectiveness in gaining a deeper understanding of Python programming techniques.

VI. CONCLUSIONS AND FUTURE WORKS

The Development of a Basic Python Programming Error Guidance Chatbot Prototype aimed to create a chatbot that assists learners in understanding Python programming while providing error guidance and query dialogues. The chatbot consists of two parts: one for general users to browse Python errors by type, and the other for administrators to manage error data. The evaluation of the prototype implementation found good usability and satisfaction among 34 samples. The chatbot prototype was created using the Dialogflow tool and has the potential to aid learners in education. This corresponds to [2] and [3] cited, the reason why users

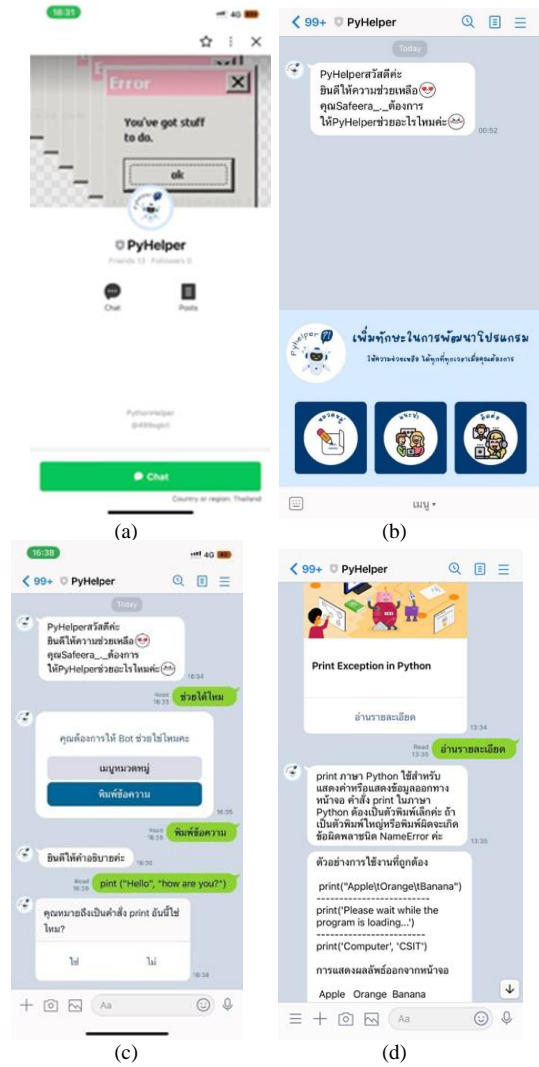
enjoy using this chatbot may be that these chatbots are designed to help them learn programming. Especially in learning Python, a very popular programming language used in IT and software development. Due to its versatility and cross-platform compatibility, Python is also an easy-to-understand programming language for beginners to learn programming. It offers a rich and powerful set of libraries for many tasks such as data processing, web applications building and development, and Game development has made Python the most popular programming language in the industry today.

Suggestions and guidelines for future development

- Contents or examples of errors should be explained to be easier to understand, for example, with pictures to add different colors of text.
- Important command descriptions should be added and frequently encountered that are essential to learning more Python language.
- If the user types a search for any word that does not appear in the chatbot, the chatbot should add the search term automatically.

These are some suggestions and guidelines for future development of the chatbot:

- Provide easier-to-understand explanations of error examples, such as using pictures and different text colors.
- Add important command descriptions that are frequently encountered and essential to learning more about Python.
- Automatically add search terms if the user types a word that does not appear in the chatbot.
- Develop advanced web applications to link with Richmenu and provide a more attractive screen to the user.



(a) Chatbot profile screen (b) chat start screen (c)Message typing screen to chat with bots (d) Error description screen and bot suggestions

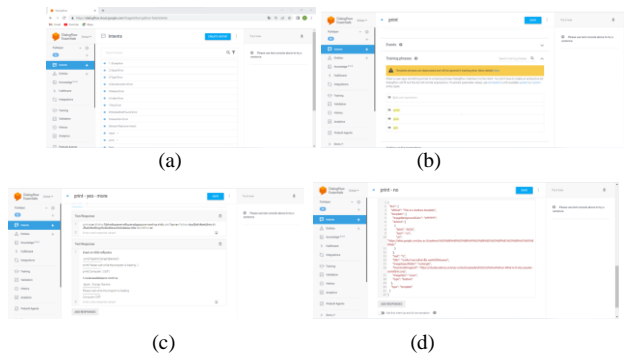


Figure 4. a) The Intent screen combines questions and responses based on boundary rules. (b) The screen for adding text, question sentences according to keywords has been defined. c) Reply message data deletion screen d) Text editing screen

REFERENCES

- [1] Eriksen, C., Rasmussen, M. H., & Høegh, R. T. (2022). Chatbots in Computer Science Education: A Literature Review. In Proceedings of the 54th Hawaii International Conference on System Sciences.
- [2] Kaur, H., & Gupta, A. (2022). Chatbot based learning for programming concepts. In 2022 International Conference on Inventive Research in Computing Applications (pp. 813-816). IEEE.
- [3] Kumar, N., Agarwal, A., & Verma, N. (2022). Design and Development of a Chatbot for Enhancing Learning of Programming Concepts. In 2022 International Conference on Computational Intelligence in Data Science (pp. 27-34). Springer, Singapore.
- [4] Chatterjee, S., & Pal, R. (2022). A Chatbot Based Intelligent Tutoring System for Enhancing Learning of Basic Programming Concepts. In Proceedings of the International Conference on

Intelligent Human Computer Interaction (pp. 229-235). Springer, Singapore.

- [5] Vajrapu, A., & Sirisha, K. (2022). Programming Assistant Chatbot using Natural Language Processing Techniques. In 2022

International Conference on Communication, Computing and Electronics Systems (ICCCES) (pp. 1-4). IEEE.

- [6] Mindphp.com (online). 9 Types of Errors in Python. Available from <https://www.mindphp.com/บทความออนไลน์/83-python/7886-specifying-type-of-error.html>. (Accessed: 1 Aug 2022).