Faculty of Computers and Data Science

Intelligent Systems Department

Knowledge-Based AI Project

Python Programming Education for Children

Team 16

Team Members

Name	ID
Ahmed Mostafa AbdelRahman	20221372883
Mazen Gaber Mahmoud	20221372110
Aya Abdelmonem Mohammed	20221462478
Doaa Samir El-Sayed	2103114

Contents

Introduction 4
Problem Statement 4
Solution Approach 5
Technical Implementation 6
    Data Collection 6
    Text Preprocessing 7
    Vector Embedding 8
    Vector Database 8
    Question Answering System 9
Results and Examples 10
    Example 1: "what is While loop in python?" 10
    Example 2: "what is if conditions in Python?" 11
    Example 3: "what is variable meaning in python?" 11
    Example 4: "what is Types of Numbers in Python?" 12
Conclusion 13

Introduction

Digital literacy and programming skills are becoming essential for the next generation. However, most programming resources use technical language that can overwhelm young learners aged 8-12. This project addresses this gap by developing a child-friendly Python programming assistant that explains coding concepts in simple, accessible ways.

Problem Statement

The primary challenges in teaching programming to children include:

- Technical terminology that is difficult for children to understand
- Lack of age-appropriate explanations for programming concepts
- Need for relatable examples that connect to children's experiences
- Limited resources that maintain children's interest while building their confidence

Solution Approach

Our solution implements a Retrieval-Augmented Generation (RAG) system specifically designed for children's Python education. The system:

1. Retrieves relevant information from a curated database of Python programming content
2. Processes this information to match the knowledge level of children
3. Generates age-appropriate explanations with simple examples
4. Uses encouraging and positive language to build confidence

The RAG approach ensures that responses are factually accurate while being presented in a way that is accessible to young learners.

Technical Implementation

Our implementation follows a systematic approach through several key phases:

Data Collection

We collected data from reliable Python programming resources using web scraping techniques:

- **Sources**: Programming tutorials from programiz.com and the official Python documentation
- **Content Types**: Both HTML pages and PDF documents
- **Scope**: Covered fundamental Python concepts including variables, data types, conditional statements, and loops
- **Volume**: Gathered 50,726 words across 5 source documents

The web scraping process used requests and BeautifulSoup libraries, with custom handlers for both HTML and PDF content. Error handling, throttling, and metadata extraction were implemented to ensure high quality data collection.

Text Preprocessing

1. **Cleaning**: Removal of irrelevant content, standardization of whitespace, and normalization of text
2. **Chunking**: Implementation of a recursive text splitter that divides content into manageable segments

Parameters:

- Chunk size: 500 words
- Chunk overlap: 100 words (to maintain context between chunks)
- Progressively smaller separators (paragraphs, sentences, words) to preserve meaning

Output:
1,064 text chunks with associated metadata (source URL, title, word count)

The recursive text splitting approach ensures that chunks maintain semantic coherence while being suitable for embedding and retrieval.

Vector Embedding

Text chunks were converted to vector representations

- **Model**: SentenceTransformer (all-MiniLM-L6-v2)
- **Vector Size**: 384 dimensions per chunk

We batched encoding with progress tracking then we made Sample verification of embeddings to ensure proper encoding

These embeddings capture the semantic meaning of each text chunk, enabling the system to find relevant content based on similarity to user questions.

Vector Database

Embeddings were stored in a vector database for efficient retrieval:

1. **Database**: ChromaDB (in-memory vector store)
2. **Collection**: Created with metadata including description and embedding function
3. **Storage**: Each document chunk stored with:

- Unique ID
- Full text content
- Metadata (URL, title, chunk ID, word count)
- Vector embedding

4. **Retrieval Function**: Implemented to retrieve the most semantically similar chunks to a query

The vector database enables fast semantic search, ranking results by relevance to the user's question.

Question Answering System

The complete RAG system integrates retrieval with answer generation:

1. **Query Processing**: Embedding user questions using the same model as the document chunks
2. **Context Retrieval**: Finding the top 3-5 most relevant text chunks
3. **Context Formatting**: Structuring retrieved information with appropriate source attribution
4. **Answer Generation**:

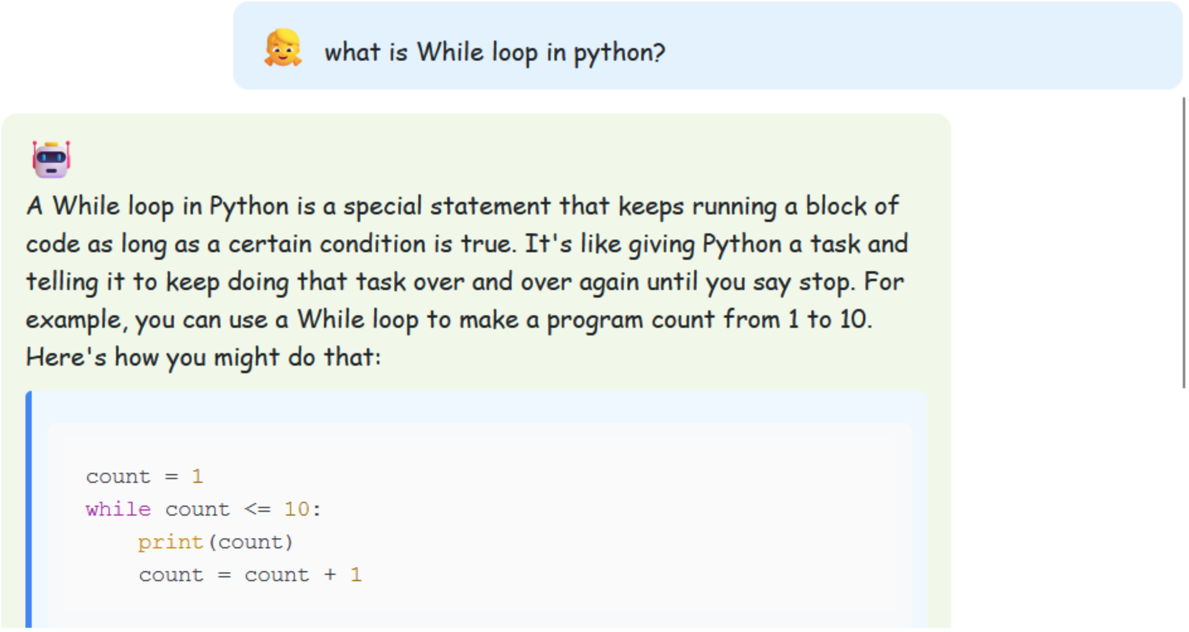
- **API**: Mistral AI API (mistral-small model)
- **System Prompt**: Custom instructions for child-friendly responses
- **Parameters**: Temperature of 0.5, max tokens of 500
- **Fallback**: Graceful handling of API failures

The system prompt specifically instructs the model to:

- Provide simple, easy-to-understand explanations
- Use encouraging and positive language
- Base answers only on the provided context
- Include simple examples when appropriate

Results and Examples

Example 1: "what is While loop in python?"

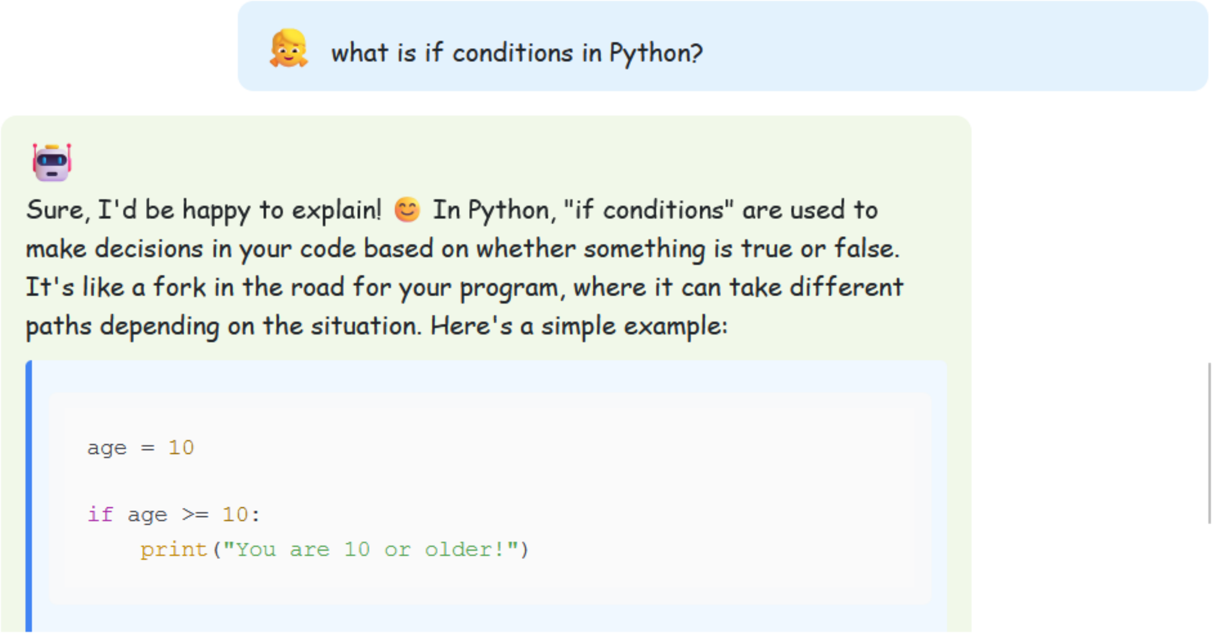


what is While loop in python?

A While loop in Python is a special statement that keeps running a block of code as long as a certain condition is true. It's like giving Python a task and telling it to keep doing that task over and over again until you say stop. For example, you can use a While loop to make a program count from 1 to 10. Here's how you might do that:

```
count = 1
while count <= 10:
    print(count)
    count = count + 1
```

Example 2: "what is if conditions in Python?"



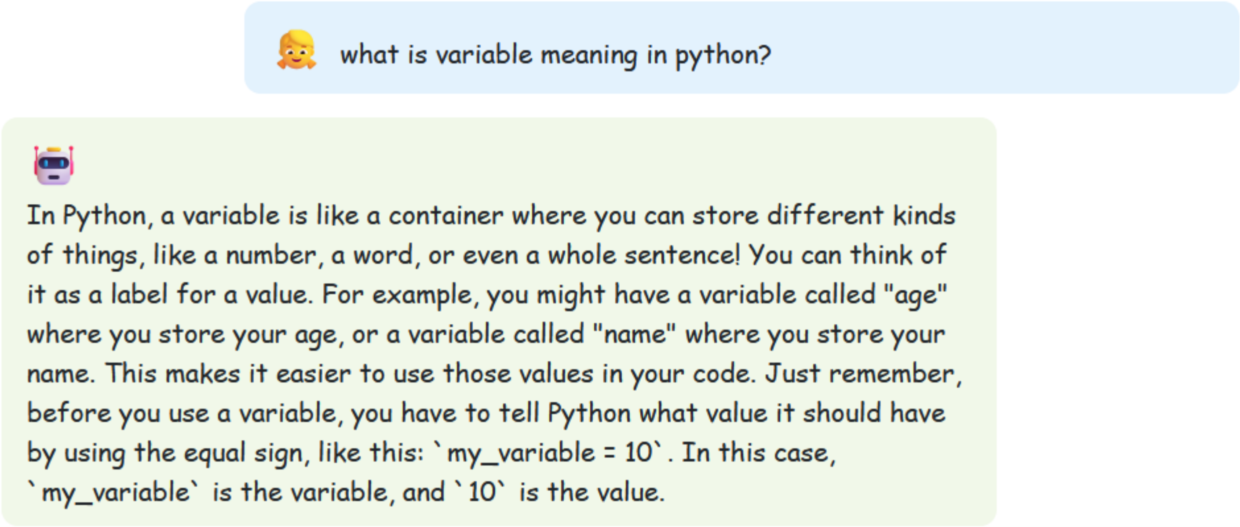
what is if conditions in Python?

Sure, I'd be happy to explain! 😊 In Python, "if conditions" are used to make decisions in your code based on whether something is true or false. It's like a fork in the road for your program, where it can take different paths depending on the situation. Here's a simple example:

```
age = 10

if age >= 10:
    print("You are 10 or older!")
```

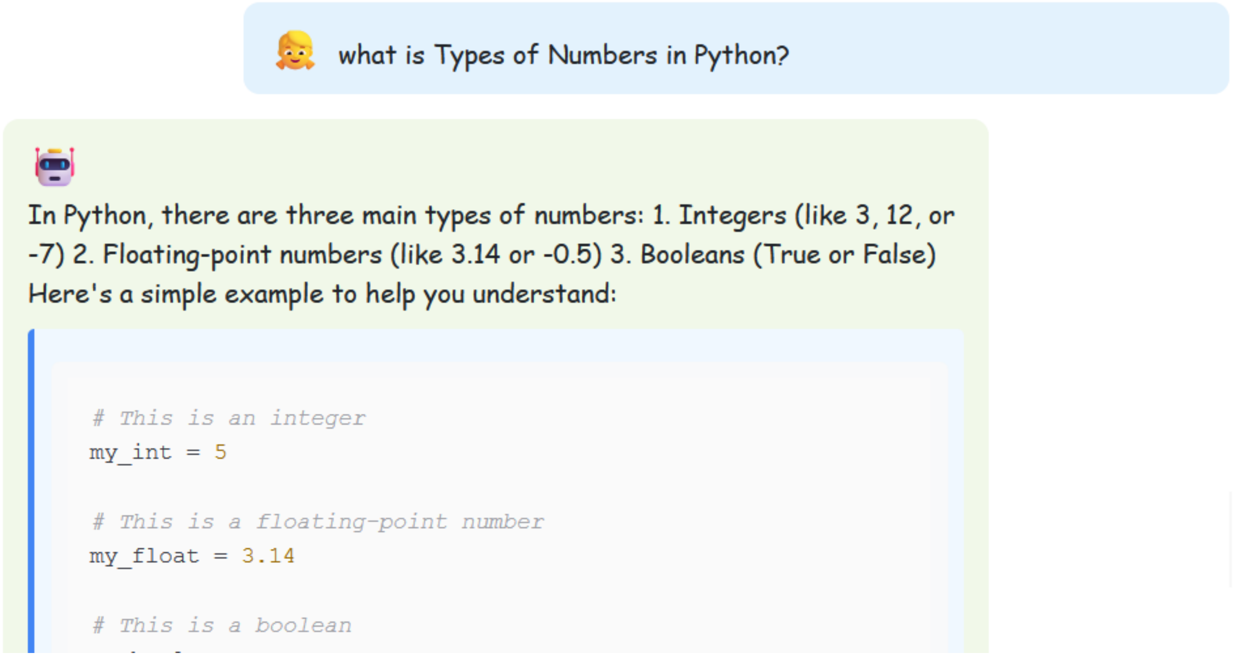
Example 3: "what is variable meaning in python?"



what is variable meaning in python?

In Python, a variable is like a container where you can store different kinds of things, like a number, a word, or even a whole sentence! You can think of it as a label for a value. For example, you might have a variable called "age" where you store your age, or a variable called "name" where you store your name. This makes it easier to use those values in your code. Just remember, before you use a variable, you have to tell Python what value it should have by using the equal sign, like this: `my_variable = 10`. In this case, `my_variable` is the variable, and `10` is the value.

Example 4: "what is Types of Numbers in Python?"



what is Types of Numbers in Python?

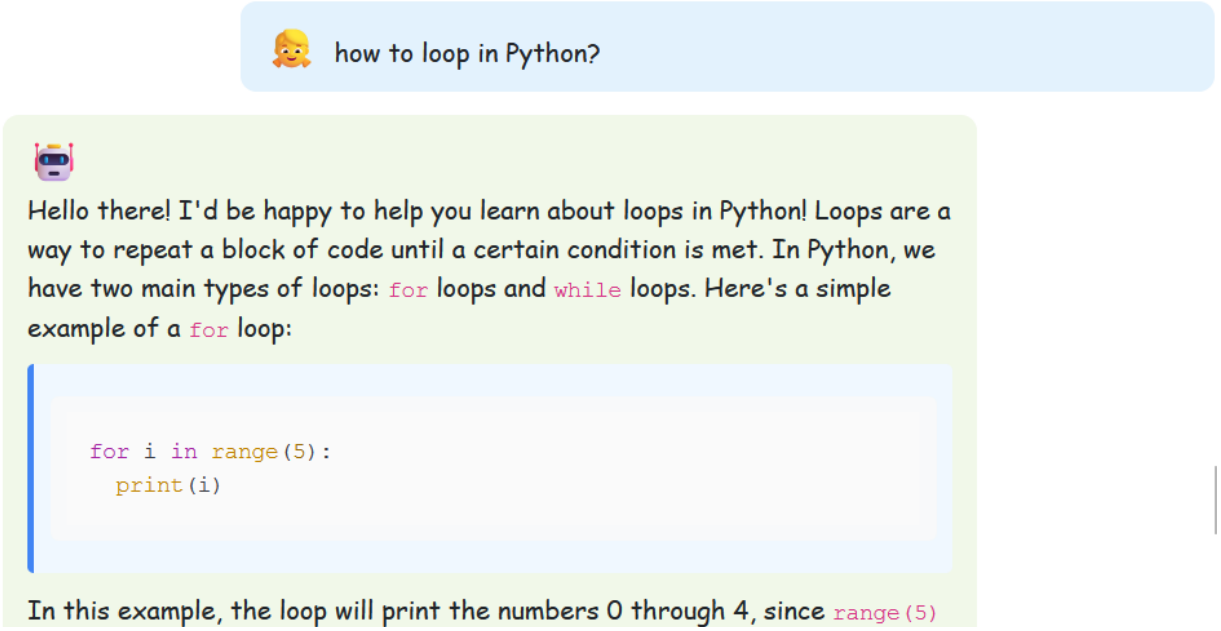
In Python, there are three main types of numbers: 1. Integers (like 3, 12, or -7) 2. Floating-point numbers (like 3.14 or -0.5) 3. Booleans (True or False) Here's a simple example to help you understand:

```
# This is an integer
my_int = 5

# This is a floating-point number
my_float = 3.14

# This is a boolean
```

Example 5: "how to loop in Python?"



how to loop in Python?

Hello there! I'd be happy to help you learn about loops in Python! Loops are a way to repeat a block of code until a certain condition is met. In Python, we have two main types of loops: `for` loops and `while` loops. Here's a simple example of a `for` loop:

```
for i in range(5):
  print(i)
```

In this example, the loop will print the numbers 0 through 4, since `range(5)`

Conclusion

The Python Programming Education for Children project successfully achieves its goal of making programming more accessible to young learners. By combining web scraping, natural language processing, and retrieval-augmented generation, we have created a system that:

1. Translates complex programming concepts into child-friendly explanations
2. Provides concrete examples that help children understand abstract concepts
3. Uses encouraging language that builds confidence and motivation
4. Delivers accurate information based on reliable programming resources