

Lab 02: Creating a Chatbot

Course: Programming for Data Science

Professor: Dr. Carlos J. Costa

September 18, 2026

Objective

In this lab, you will use basic Python programming concepts to progressively build a simple text-based chatbot.

You will practise:

- variables and input from the user;
- type conversion;
- arithmetic operations;
- `if` statements;
- `while` loops;
- functions;
- function parameters;
- combining these concepts into a small interactive program.

Important: Complete the exercises in order. Each exercise introduces a concept that will be used in the next one.

1 Getting Information from the User

Ask the user to enter their **name** and **age**.

Then calculate the user's approximate **birth year** using the current year, 2026.

The program should display a message similar to:

Hello Carlos. You were approximately born in 2000.

Hint: `input()` returns text. You will need to convert the age to an integer before performing the calculation.

Write your code here

2 A Simple Conversation

Create a program that asks the user to enter a text.

The program should:

1. ask the user for a message;
2. always answer **yes**;
3. continue asking for messages;
4. stop the conversation when the user enters **bye**.

The conversation should look approximately like this:

You: Do you like Python?

Chatbot: yes

You: Is Python useful for data science?

Chatbot: yes

You: bye

Chatbot: Goodbye!

Hint: Use a `while` loop. Check the user's input before deciding whether to continue.

```
# Write your code here
```

3 Creating a Greeting Function

Create a function called `greeting()` that displays a greeting message for the user.

For example, the function could display:

Hello! I am your chatbot.

Test the function after defining it.

```
# Define the greeting() function
```

```
# Test the function
```

4 Using the Greeting Function

Modify the conversation from Exercise 2 so that the chatbot calls the `greeting()` function **at the beginning of the conversation**.

The interaction should start with something similar to:

Chatbot: Hello! I am your chatbot.

You: Hello

Chatbot: yes

The chatbot should still stop when the user enters **bye**.

```
# Write your code here
```

5 Build Your Own Chatbot

Using the previous exercises as a starting point, create a simple chatbot.

Your chatbot must:

1. call a greeting function when the conversation starts;
2. ask the user for messages;
3. continue the conversation using a `while` loop;
4. recognise `bye` and close the conversation;
5. use at least **one additional function** besides `greeting()`;
6. provide at least **three different responses**, depending on what the user types.

For example, you could make the chatbot respond differently to:

- `hello / hi`;
- questions containing `python`;
- questions containing `data`;
- `bye`.

Do not use artificial intelligence or an external API. The objective is to create the chatbot using the Python programming concepts introduced in this lab.

Example interaction

Chatbot: Hello! I am your chatbot.

You: hi

Chatbot: Hello! Nice to meet you.

You: Tell me something about Python.

Chatbot: Python is widely used in data science.

You: What about data?

Chatbot: Data is an important part of data science.

You: bye

Chatbot: Goodbye!

The responses and topic of the chatbot are your choice.

Create your chatbot here

6 Challenge: Make the Chatbot More Flexible

Improve your chatbot so that it is less dependent on the exact text entered by the user.

For example, make these inputs produce the same type of response:

Hello

hello

HELLO

Hi

Hint: Investigate the `lower()` method for strings.

You may also add new topics and responses.

Improve your chatbot here

7 Reflection

Answer the following questions in your own words:

1. Why is a `while` loop useful for a chatbot?
2. What is the advantage of using functions in the chatbot?
3. What is the difference between `input()` and `print()`?
4. Why do we need to convert the user's age to an integer in Exercise 1?
5. What are the main limitations of a rule-based chatbot such as the one created in this lab?

Summary

In this lab, you combined fundamental Python concepts to create an interactive program. The chatbot is **rule-based**: its behaviour is determined by conditions written in the program rather than by an AI model.

The same programming concepts—input, variables, conditions, loops, and functions—are fundamental building blocks for more complex data science and artificial intelligence applications.