CHAPTER 3



# Getting Your Hands Dirty: Conversation Module

In this chapter, you learn how to implement a conversation module to make Melissa understand what you are saying, with the help of a Python program that implements keyword-recognition techniques. You refine the code of the program to make it more efficient, so that you can have a general conversation with Melissa and ask questions like, "How are you?" and "Who are you?"

You have reached the step of building a virtual assistant that involves designing a logic engine. Melissa is basically a parrot right now, repeating what you say. This assistant needs to be more than that; it needs to understand what you say. In a quest to make Melissa smart, let's design a conversation module.

Before you learn how to implement this module in Python, let's revisit the code skeleton from Chapter 1 and see how you build and add components of the logic engine, keeping the different modules isolated from each other. You have already incorporated the STT and TTS in the code skeleton, so in this chapter you immediately implement the code you develop into the project instead of prototyping.

## Logic Engine Design

`main.py` is the STT engine of your software, and it is also the entry point to your program. You need `main.py` to direct user queries to its logic engine, which you code in the `brain.py` file. The `brain.py` file will contain a ladder of `if/else` clauses to determine what the user wants to say. If there is a pattern match with one of the statements, `brain.py` call the corresponding module.

Figure 3-1 shows the control flow of the program. This will be similar for all the modules you develop for Melissa in future chapters. The difference will be that some other module is called by `brain.py` instead of `general_conversations.py`.

The `GreyMatter` package will hold logic-engine modules that you build to make Melissa smarter in the future, such as a weather module, opening a web site, playing music, and so on. The `GreyMatter` package also contains the `general_conversations.py` file.



T. Pant, *Building a Virtual Assistant for Raspberry Pi*, DOI 10.1007/978-1-4842-2167-9_3