Create a file named `define_subject.py` in the `GreyMatter` folder, and enter the following code:

```
import re

import wikipedia

from SenseCells.tts import tts

def define_subject(speech_text):
    words_of_message = speech_text.split()
    words_of_message.remove('define')
    cleaned_message = ' '.join(words_of_message)

    try:
        wiki_data = wikipedia.summary(cleaned_message, sentences=5)

        regEx = re.compile(r'([^\(]*)\([^\)]*\) *(.*)')
        m = regEx.match(wiki_data)
        while m:
            wiki_data = m.group(1) + m.group(2)
            m = regEx.match(wiki_data)

        wiki_data = wiki_data.replace("'", "")
        tts(wiki_data)
    except wikipedia.exceptions.DisambiguationError as e:
        tts('Can you please be more specific? You may choose something from
        the following.')
        print("Can you please be more specific? You may choose something
        from the following.; {0}".format(e))
```

This code imports the regular-expressions module named `re` and the `wikipedia` module that you installed via `pip`. The best approach to understand this code is working through a sample case.

Suppose the user gives the command "define tanay pant." It is passed as an argument to the `define_subject()` function. The string is split into an array of words, from which the word *define* is removed. This leaves you with the following two words in the array: *tanay* and *pant*. These two words form the subject that needs defining. This new array of words is rejoined and assigned to the `cleaned_message` variable. It then seeks a summary from `wikipedia` via the `wikipedia.summary()` function. You specify two arguments: the subject—that is, the `cleaned_message` variable—and the number of sentences the summary should contain.

The next statement consists of a regular-expression pattern match that removes anything in braces (braces inclusive) from the summary and then recombines the summary. You do this because braces will mess with the `tts()` function. It then removes all the apostrophe (`'`) characters from the summary, because they also interfere with `tts()`. Finally, the result is spoken via the `tts()` function.