Now it's time to edit `brain.py`; make the following edits and additions. The first change is to the `import` statement:

```
from GreyMatter import define_subject, tell_time, general_conversations,
play_music, weather, connect_proxy, open_firefox, sleep, business_news_reader
```

Edit the declaration of the `brain()` function to make it look like the following:

```
def brain(name, speech_text, music_path, city_name, city_code, proxy_
username, proxy_password):
```

The last step is to add a code snippet to call the two functions in the file's logic-handling `if/else` clause:

```
elif check_message(['play', 'music']) or check_message(['music']):
    play_music.play_random(music_path)

elif check_message(['play']):
    play_music.play_specific_music(speech_text, music_path)
```

So far, so good! Note that you purposely put the `play_random()` function first, because it recognizes the call made to it via the *play* and *music* keywords. If you put just the *play* keyword first, as in the `play_specific_music()` function, then even if you want to hear a random track, the module would split the query and completely mess it up, resulting in "track not found" and hence an error. The first clause in an `if/else` ladder takes the first priority.

Play Party Mix!

This is an exercise for you: create another function in the `play_music` module that makes a list of all the MP3 files, shuffles them, and then plays them one by one. This feature should be invoked when keywords such as *party* and *mix* or *party* and *time* are present.

I encourage you to try to implement this feature yourself before looking at the following solution. Doing so will improve your understanding of this software and help you scale the software in the future.

First, make a function named `play_shuffle()` in `play_music.py`. Type in the following code:

```
def play_shuffle(music_path):
    try:
        music_listing = mp3gen(music_path)
        random.shuffle(music_listing)
        for i in range(0, len(music_listing)):
            music_player(music_listing[i])
    except IndexError as e:
        tts('No music files found.')
        print("No music files found: {0}".format(e))
```