



**Figure 7-3.** Tweet posted on Twitter

This code works like a charm. Also notice that it has capitalized “Hello” because I used the `capitalize()` function on the `cleaned_message()` variable.

## Exercises

Here are a few tasks for you to accomplish by yourself in order to extend the functionality of Melissa’s Twitter module and make it more useful. First, add a function in the `twitter_interaction` file to search for a person’s username and dictate the details about them. Another interesting feature you can add is to have Twitter search for tweets on a particular topic or subject. You should also be able to hear your own tweets as well as the tweets of any other user. And a Twitter application is never complete in the truest sense if it cannot send direct messages to your followers.

Adding these functionalities to the `twitter_interaction.py` file will improve your understanding of the `tweepy` module and let you voice-control Twitter like a pro. If you successfully implement these features, be sure to send a pull request to the `Melissa-AI/Melissa-Core` repository on GitHub to get your code merged into the official repository.

## Building the Imgur Module

Imgur is an image-upload web site that provides an API and a Python module for uploading images and sharing them with your friends. Uploading images is a boring task—you have to click the Upload File button, select the file, and then click the Upload button. Drag-and-drop uploads certainly make the task easier, but you still have to find the file and use the mouse to perform the drag-and-drop operation.

It would definitely make things easier for you if Melissa could do all this for you, wouldn’t it? Also, it would be even better if the URLs of the images you upload to Imgur are stored in a database, along with the dates. Let’s build a module that can perform these tasks and simplify life for you!