As is evident from the code, the `connect_to_proxy()` function first opens the browser, then opens the proxy URL, and then searches for the username text field and the password field. Finally, it enters the username and the password in their respective fields and presses Enter to log in the user.

You can make a similar login-automation module for any web site by first making the changes in the URL, username, and password ID, class, or name (from the source code), and changing the username and password.

Make the final edits and additions in `brain.py`:

```
from GreyMatter import tell_time, general_conversations, weather,
define_subject, business_news_reader, open_firefox, connect_proxy
    elif check_message(['connect', 'proxy']):
        connect_proxy.connect_to_proxy(proxy_username, proxy_password)
```

Now, giving Melissa a command like "Connect to proxy server" will cause her to automatically connect you to the proxy server.

Remember that you should never store passwords in plaintext in publicly accessible files. As I mentioned, you may want to re-create the earlier instructions and store the username and password in a database using an encryption scheme such as SHA2.

Time to Sleep, Melissa!

Let's write a short module to shut down the software and ask Melissa to sleep. Type the following code for `sleep.py` in the `GreyMatter` folder:

```
from SenseCells.tts import tts

def go_to_sleep():
    tts('Goodbye! Have a great day!')
    quit()
```

Time to make the edits in `brain.py`:

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

    elif check_message(['sleep']):
        sleep.go_to_sleep()
```

Now, saying something like "Time to sleep, Melissa!" or "Sleep!" will shut down the software and exit from the Python script.

You should commit your code after building a feature or completing the lessons in any chapter. This will help you to go back to your previous state (the last time you committed) if you accidently delete something or mess something up. You can go back to the previously committed state by entering the following command on your terminal:

```
$ git reset --hard
```