This lets you open Firefox by giving commands like "Open Firefox."

You can do a host of pretty interesting things with Selenium. For example, you can automate logging in to a web site like Facebook so that when you give the command to Melissa, she opens a browser, opens the web site, and then logs in to the web site using your credentials.

Some Internet connections need the user to log in to proxy portals, also known as *captive portals*, to access the Internet. You can also automate this type of login. Let's write a module for performing this functionality.

Store the proxy username and password in the `profile.yaml` file:

```
proxy_username:
  Something
proxy_password:
  Something
```

I would like to point out that saving passwords in plaintext and in publicly accessible files is a very bad idea. You may wish to save the password in a database in an encrypted format. However, for the sake of simplicity, keep it in `profile.yaml` for now.

Make the following additions to `main.py`:

```
proxy_username = profile_data['proxy_username']
proxy_password = profile_data['proxy_password']
```

Edit the call to the `brain()` function in `main()` as follows (and also in the function declaration in `brain.py`):

```
brain(name, speech_text city_name, city_code, proxy_username, proxy_password)
```

Now, create a `connect_proxy.py` file in the `GreyMatter` folder, and type the following code in it:

```
from selenium import webdriver

from SenseCells.tts import tts

def connect_to_proxy(proxy_username, proxy_password):
    tts("Connecting to proxy server.")
    browser = webdriver.Firefox()
    browser.get('http://10.1.1.9:8090/httpclient.html')

    id_number = browser.find_element_by_name('username')
    password = browser.find_element_by_name('password')

    id_number.send_keys(proxy_username)
    password.send_keys(proxy_password)

    browser.find_element_by_name('btnSubmit').click()
```