



POST-PUBLICATION ADDITION: You can install all of the needed modules (with the versions used in this book) by installing the `automateboringstuff` module with Pip. Run `pip install --user automateboringstuff` from the Command Prompt or Terminal window.

A INSTALLING THIRD-PARTY MODULES



Many developers have written their own modules, extending Python's capabilities beyond what is provided by the standard library of modules packaged with Python. The primary way to install third-party modules is to use Python's `pip` tool. This tool securely downloads and installs Python modules onto your computer from <https://pypi.python.org/>, the website of the Python Software Foundation. PyPI, or the Python Package Index, is a sort of free app store for Python modules.

THE PIP TOOL

While `pip` comes automatically installed with Python 3.4 and later on Windows and macOS, you may have to install it separately on Linux. You can see whether `pip` is already installed on Linux by running `which pip3` in a Terminal window. If it's installed, you'll see the location of `pip3` displayed. Otherwise, nothing will display. To install `pip3` on Ubuntu or Debian Linux, open a new Terminal window and enter `sudo apt-get install python3-pip`. To install `pip3` on Fedora Linux, enter `sudo yum install python3-pip` into a Terminal window. You'll need to enter the administrator password for your computer.

The pip tool is run from a *terminal* (also called *command line*) window, not from Python’s interactive shell. On Windows, run the “Command Prompt” program from the Start menu. On macOS, run Terminal from Spotlight. On Ubuntu Linux, run Terminal from Ubuntu Dash or press CTRL-ALT-T.

If pip’s folder is not listed in the `PATH` environment variable, you may have to change directories in the terminal window with the `cd` command before running pip. If you need to find out your username, run `echo %USERNAME%` on Windows or `whoami` on macOS and Linux. Then run `cd pip folder`, where pip’s folder is `C:\Users\<USERNAME>\AppData\Local\Programs\Python\Python37\Scripts` on Windows. On macOS, it is in `/Library/Frameworks/Python.framework/Versions/3.7/bin/`. On Linux, it is in `/home/<USERNAME>/.local/bin/`. Then you’ll be in the right folder to run the pip tool.

INSTALLING THIRD-PARTY MODULES

The executable file for the pip tool is called *pip* on Windows and *pip3* on macOS and Linux. From the command line, you pass it the command `install` followed by the name of the module you want to install. For example, on Windows you would enter `pip install --user MODULE`, where *MODULE* is the name of the module.

Because future changes to these third-party modules may be backward incompatible, I recommend that you install the exact versions used in this book, as given later in this section. You can add `-U MODULE==VERSION` to the end of the module name to install a particular version. Note that there are two equal signs in this command line option. For example, `pip install --user -U send2trash==1.5.0` installs version 1.5.0 of the `send2trash` module.

You can install all of the modules covered in this book by downloading the “requirements” files for your operating system from <https://nostarch.com/automatestuff2/> and running one of the following commands:

- On Windows:

```
pip install --user -r automate-win-requirements.txt --user
```

- On macOS:

```
pip3 install --user -r automate-mac-requirements.txt --user
```

- On Linux:

```
pip3 install --user -r automate-linux-requirements.txt --user
```

The following list contains the third-party modules used in this book along with their versions. You can enter these commands separately if you only want to install a few of these modules on your computer.

- `pip install --user send2trash==1.5.0`
- `pip install --user requests==2.21.0`
- `pip install --user beautifulsoup4==4.7.1`
- `pip install --user selenium==3.141.0`
- `pip install --user openpyxl==2.6.1`
- `pip install --user PyPDF2==1.26.0`
- `pip install --user python-docx==0.8.10` (install `python-docx`, not `docx`)
- `pip install --user imapclient==2.1.0`
- `pip install --user pyzmail36==1.0.4`
- `pip install --user twilio`
- `pip install --user ezgmail`
- `pip install --user ezsheets`
- `pip install --user pillow==9.2.0`
- `pip install --user pyobjc-framework-Quartz==5.2` (on macOS only)
- `pip install --user pyobjc-core==5.2` (on macOS only)
- `pip install --user pyobjc==5.2` (on macOS only)
- `pip install --user python3-xlib==0.15` (on Linux only)
- `pip install --user pyautogui`

NOTE

For macOS users: The `pyobjc` module can take 20 minutes or longer to install, so don't be alarmed if it takes a while. You should also install the `pyobjc-core` module first, which will reduce the overall installation time.

After installing a module, you can test that it installed successfully by running `import ModuleName` in the interactive shell. If no error messages are displayed, you can assume the module was installed successfully.

If you already have the module installed but would like to upgrade it to the latest version available on PyPI, run `pip install --user -U MODULE` (or `pip3 install --user -U MODULE` on macOS and Linux). The `--user` option installs the module in your home directory. This

avoids potential permissions errors you might encounter when trying to install for all users.

The latest versions of the Selenium and OpenPyXL modules tend to have changes that are backward incompatible with the versions used in this book. On the other hand, the Twilio, EZGmail, and EZSheets modules interact with online services, and you might be required to install the latest version of these modules with the `pip install --user -U` command.

WARNING

The first edition of this book suggested using the `sudo` command if you encountered permission errors while running `pip`: `sudo pip install module`. This is a bad practice, as it installs modules to the Python installation used by your operating system. Your operating system may run Python scripts to carry out system-related tasks, and if you install modules to this Python installation that conflict with its existing modules, you could create hard-to-fix bugs. Never use `sudo` when installing Python modules.

INSTALLING MODULES FOR THE MU EDITOR

The Mu editor has its own Python environment, separate from the one that typical Python installations have. To install modules so that you can use them in scripts launched by Mu, you must bring up the Admin Panel by clicking the gear icon in the lower-right corner of the Mu editor. In the window that appears, click the Third Party Packages tab and follow the instructions for installing modules on that tab. The ability to install modules into Mu is still an early feature under development, so these instructions may change.

If you are unable to install modules using the Admin Panel, you can also open a Terminal window and run the `pip` tool specific to the Mu editor. You'll have to use `pip`'s `--target` command line option to specify Mu's module folder. On Windows, this folder is `C:\Users\<USERNAME>\AppData\Local\Mu\pkgs`. On macOS, this folder is `/Applications/mu-editor.app/Contents/Resources/app_packages`. On Linux, you don't need to enter a `--target` argument; just run the `pip3` command normally.

For example, after you download the requirements file for your operating system from <https://nostarch.com/automatestuff2/>, run the following:

- On Windows:

```
pip install -r automate-win-requirements.txt --target "C:\Users\USERNAME
\AppData\Local\Mu\pkgs"
```

- On macOS:

```
pip3 install -r automate-mac-requirements.txt --target /Applications/  
mu-editor.app/Contents/Resources/app_packages
```

- On Linux:

```
pip3 install --user -r automate-linux-requirements.txt
```

If you want to install only some of the modules, you can run the regular `pip` (or `pip3`) command and add the `--target` argument.



Read the author's other free programming books on [InventWithPython.com](https://inventwithpython.com).

Support the author with a purchase:

[Buy Direct from Publisher \(Free Ebook!\)](#) | [Buy on Amazon](#)

