Appendix B. Install Python 3

Most of the examples in this book were written and tested with Python 3.7, the most recent stable version at the time of writing. The What's New in Python page presents what was added in each version. There are many sources of Python and many ways to install a new version. In this appendix, I describe a few of these ways:

- A standard installation downloads Python from *python.org*, and adds the helper programs `pip` and `virtualenv`.
- If your work is heavily scientific, you may prefer to get Python bundled with many scientific packages from Anaconda and use its package installer `conda` instead of `pip`.

Windows doesn't have Python at all, and macOS, Linux, and Unix tend to have old versions. Until they catch up, you may need to install Python 3 yourself.

Check Your Python Version

In a terminal or terminal window, type `python -V`:

```
$ python -V
Python 3.7.2
```

Depending on your operating system, if you don't have Python or the operating system can't find it, you'll get some error message like *command not found*.

If you do have Python and it's version 2, you may want to install Python 3—either system wide, or just for yourself in a virtualenv (see "Use virtualenv", or "Install virtualenv"). In this appendix, I show how to install Python 3 system wide.

Install Standard Python

Go to the official Python download page with your web browser. It tries to guess your operating system and present the appropriate choices, but if it guesses wrong, you can use these:

- Python Releases for Windows
- Python Releases for macOS
- Python Source Releases (Linux and Unix)

You'll see a page similar to that shown in Figure B-1.

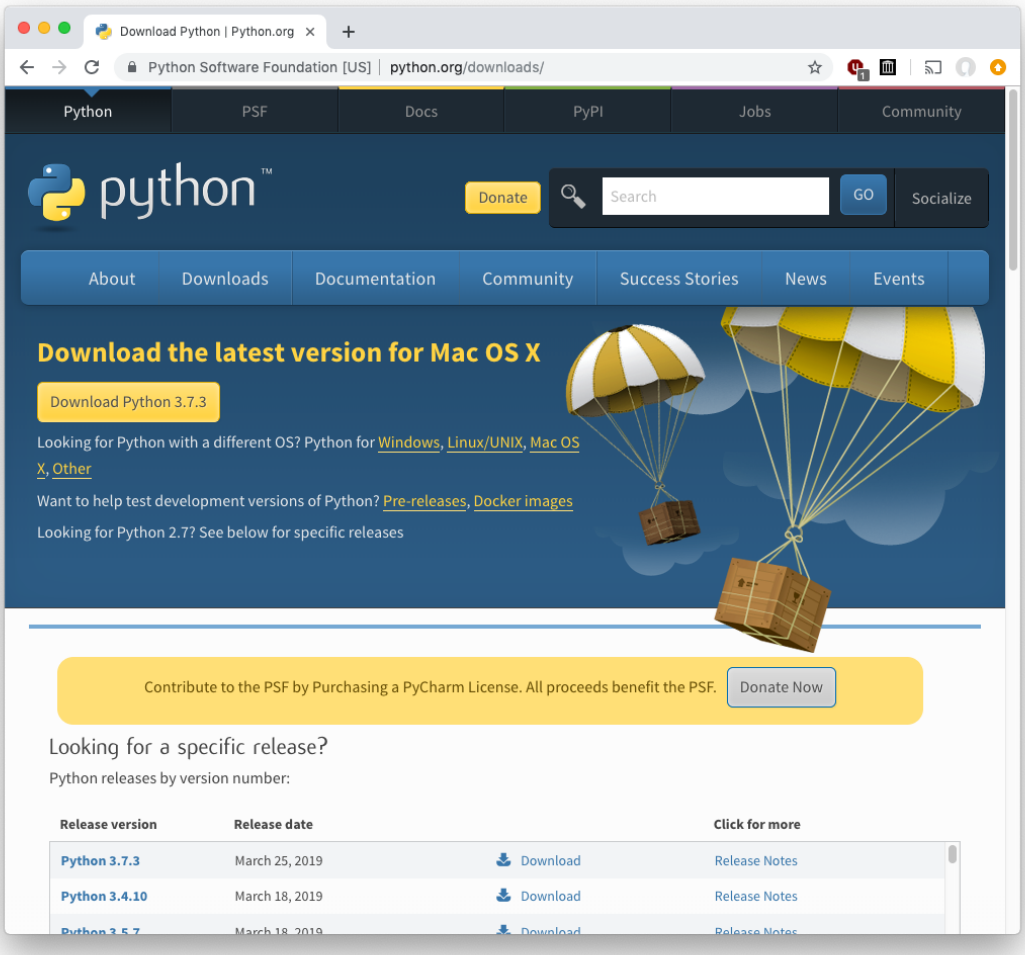

Figure B-1. Sample download page

If you click the yellow Download Python 3.7.3 button, it will download that version for your operating system. If you'd like to learn a little about it first, click the blue link text Python 3.7.3 in the first column of the table at the bottom, under Release version. This takes you to an information

page like the one shown in Figure B-2.

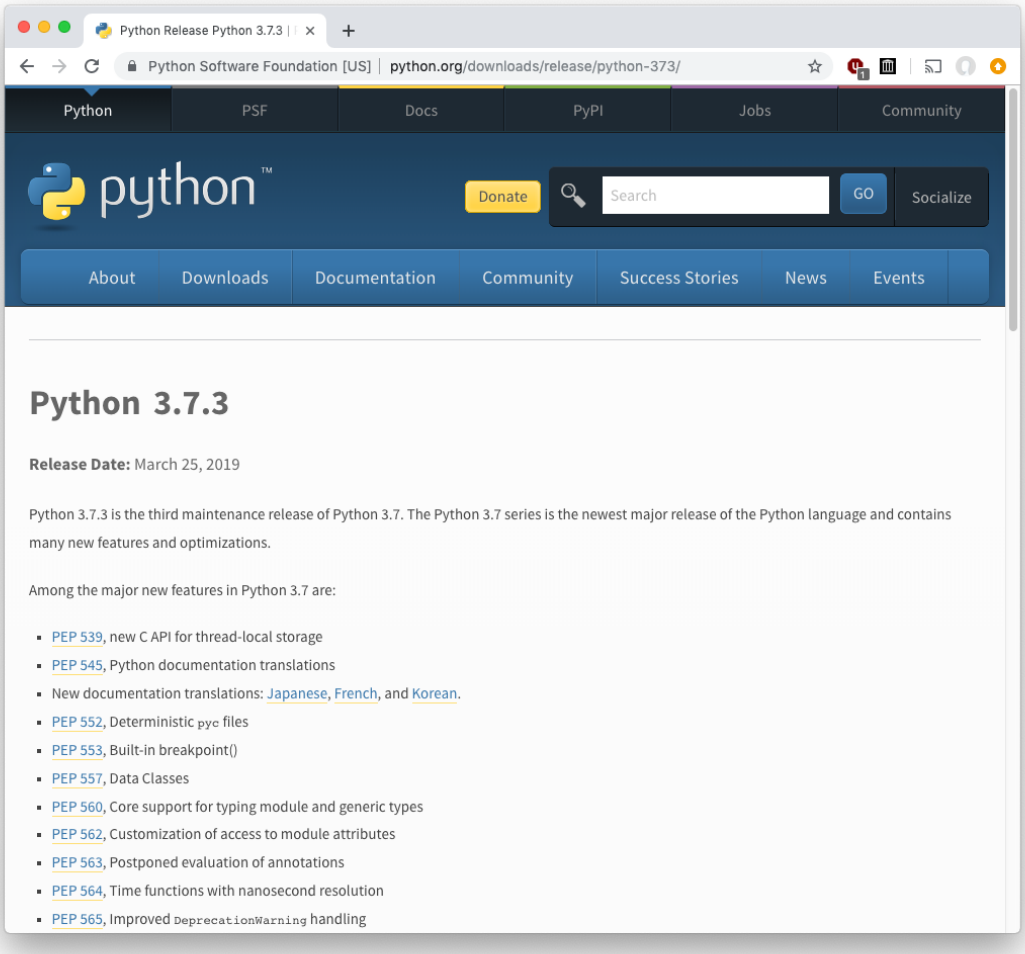

Figure B-2. Detail page for download

You need to scroll down the page to see the actual download links (Figure B-3).

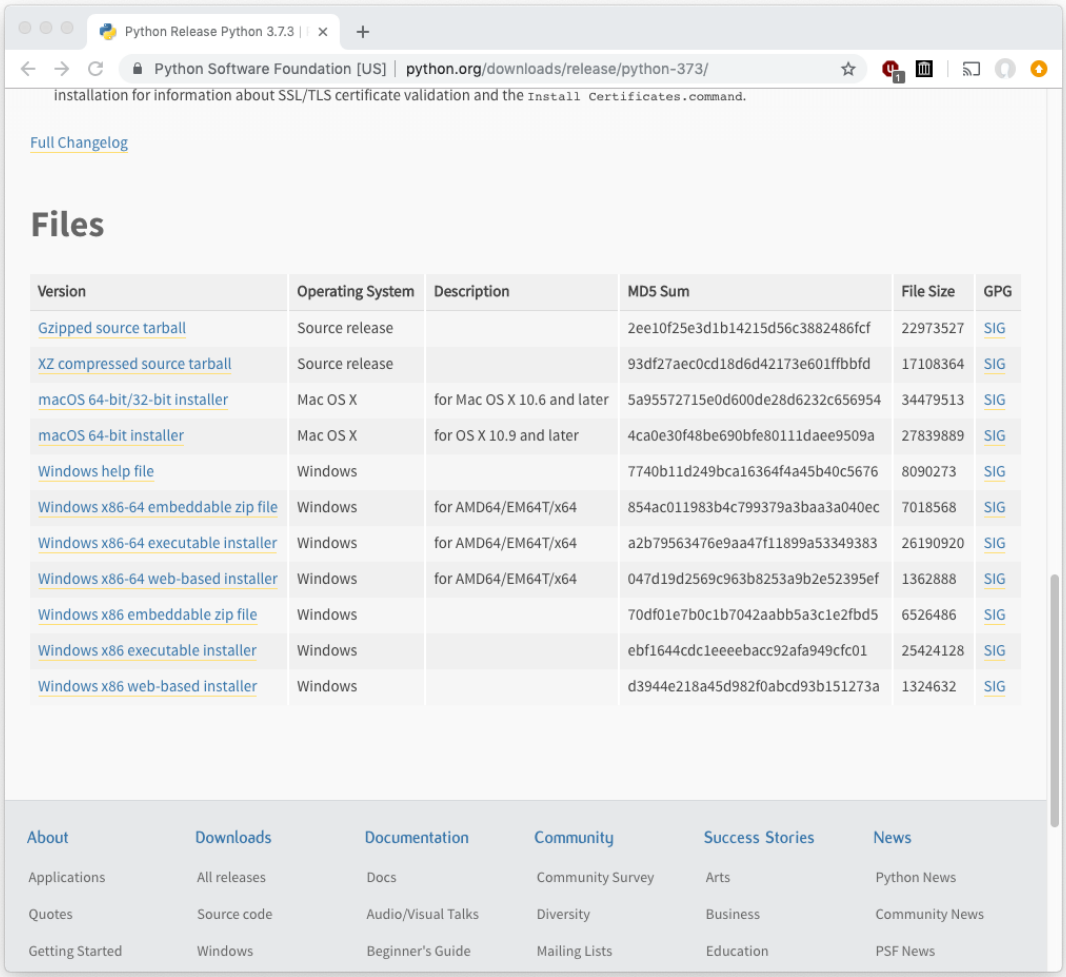

Figure B-3. Bottom of page offering downloads

macOS

Click the macOS 64-bit/32-bit installer link to download a Mac *.pkg* file. Double-click it to see an introductory dialog box (Figure B-4).

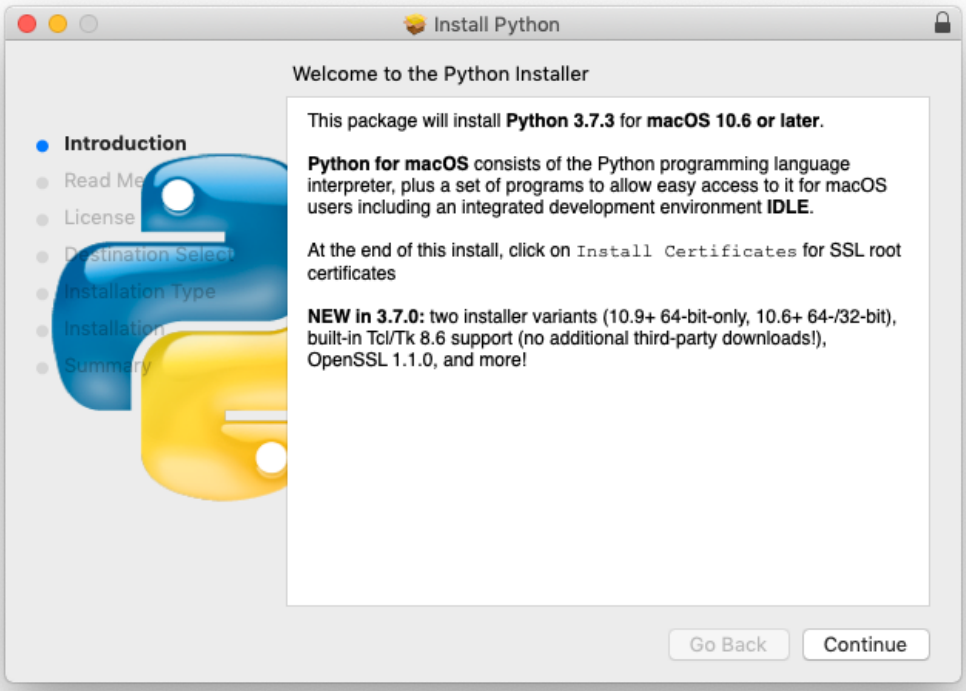

Figure B-4. Mac install dialog 1

Click Continue. You'll go through a succession of other dialog boxes.

When it's all done, you should see the dialog shown in Figure B-5.

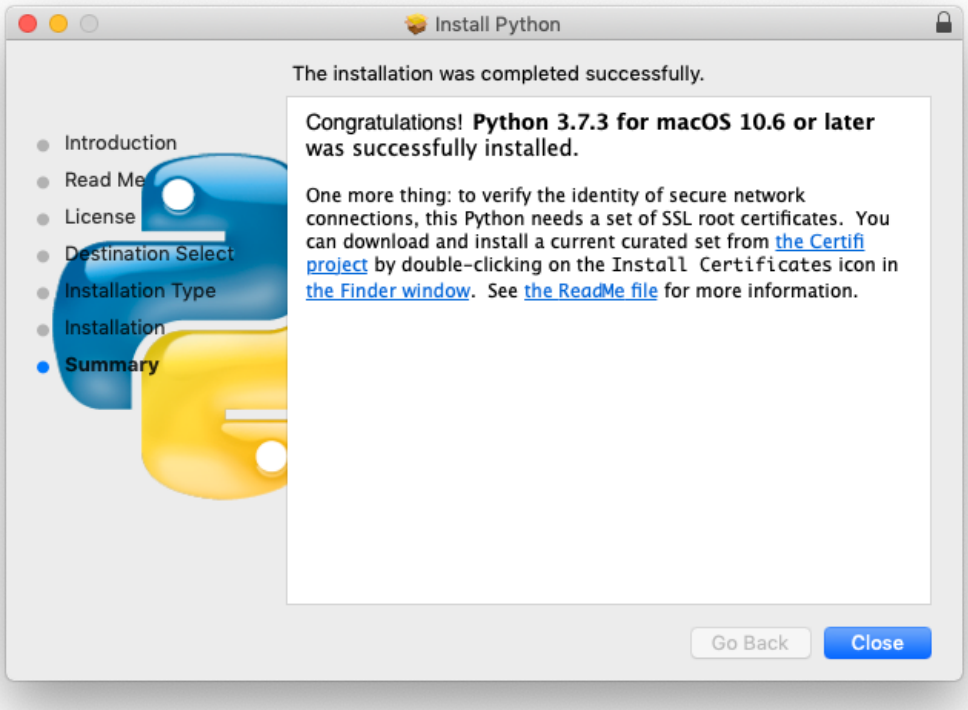

Figure B-5. Mac install dialog 9

Python 3 will be installed as */usr/local/bin/python3*, leaving any existing Python 2 on your computer unchanged.

Windows

Windows has never included Python, but recently made it easier to install. The May 2019 update for Windows 10 includes *python.exe* and *python3.exe* files. These aren't the Python interpreter, but links to a new Python 3.7 page at the Microsoft Store. You can use this link to download and install Python in the same way you get other Windows software.

Or you can download and install Python from the official Python site:

- Windows x86 MSI installer (32-bit)
- Windows x86-64 MSI installer (64-bit)

To determine whether you have a 32-bit or 64-bit version of Windows:

- Click the Start button.
- Right-click Computer.

- Click Properties and find the bit value.

Click the appropriate installer (*.msi* file). After it's downloaded, double-click it and follow the installer directions.

Linux or Unix

Linux and Unix users get a choice of compressed source formats:

- XZ compressed source tarball
- Gzipped source tarball

Download either one. Decompress it by using `tar xJ` (*.xz* file) or `tar xz` (*.tgz* file) and then run the resulting shell script.

Install the pip Package Manager

Beyond the standard Python installation, two tools are almost essential for Python development: `pip` and `virtualenv`.

The `pip` package is the most popular way to install third-party (nonstandard) Python packages. It has been annoying that such a useful tool isn't part of standard Python and that you've needed to download and install it yourself. As a friend of mine used to say, it's a cruel hazing ritual. The good news is that `pip` is a standard part of Python, starting with the 3.4 release.

If you have Python 3 but only the Python 2 version of `pip`, here's how to get the Python 3 version on Linux or macOS:

```
$ curl -O http://python-distribute.org/distribute_setup.py
$ sudo python3 distribute_setup.py
$ curl -O https://raw.github.com/pypa/pip/master/contrib/get-pip.py
$ sudo python3 get-pip.py
```

This installs `pip-3.3` in the `bin` directory of your Python 3 installation. Then, use `pip-3.3` to install third-party Python packages rather than

Python 2's `pip`.

Install virtualenv

Often used with `pip`, the `virtualenv` program is a way to install Python packages in a specified directory (folder) to avoid interactions with any preexisting system Python packages. This lets you use whatever Python goodies you want, even if you don't have permission to change the existing installation.

Some good guides to `pip` and `virtualenv` are:

- A Non-Magical Introduction to Pip and Virtualenv for Python Beginners
- The Hitchhiker's Guide to Packaging: Pip

Other Packaging Solutions

As you've seen, Python's packaging techniques vary, and none work well for every problem. The PyPA (Python Packaging Authority) is a volunteer working group (not part of the official Python development core group) that's trying to simplify Python packaging. The group wrote the Python Packaging User's Guide, which discusses problems and solutions.

The most popular tools are `pip` and `virtualenv`, and I've used these throughout this book. If they fall short for you, or if you like trying new things, here are some alternatives:

- `pipenv` combines `pip` and `virtualenv` and adds more features. See also some criticism and threaded discussion.
- `poetry` is a rival that addresses some of the problems with `pipenv`.

But the most prominent packaging alternative, especially for scientific and data-heavy applications, is `conda`. You can get it as part of the Anaconda Python distribution, which I talk about next, or by itself ("Install Anaconda's Package Manager conda").

Install Anaconda

Anaconda is an all-in-one distribution with an emphasis on science. The latest version, Anaconda3, includes Python 3.7 and its standard library as well as the R language for data science. Other goodies include libraries that we've talked about in this book: `beautifulsoup4`, `flask`, `ipython`, `matplotlib`, `nose`, `numpy`, `pandas`, `pillow`, `pip`, `scipy`, `tables`, `zmq`, and many others. It also has a cross-platform installation program called `conda`, which I get to in the next section.

To install Anaconda3, go to the download page for the Python 3 versions. Click the appropriate link for your platform (version numbers might have changed since this was written, but you can figure it out):

- The macOS installer will install everything to the *anaconda* directory under your home directory.
- For Windows, double-click the *.exe* file after it downloads.
- For Linux, choose the 32-bit version or the 64-bit version. After it has downloaded, execute it (it's a big shell script).

NOTE

Ensure that the name of the file you download starts with *Anaconda3*. If it starts with just *Anaconda*, that's the Python 2 version.

Anaconda installs everything in its own directory (*anaconda* under your home directory). This means that it won't interfere with any versions of Python that might already be on your computer. It also means that you don't need any special permission (account names like `admin` or `root`) to install it either.

Anaconda now includes more than 1,500 open source packages. Visit the Anaconda docs page and click the link for your platform and Python version.

After installing Anaconda3, you can see what Santa put on your computer

by typing the command `conda list`.

Installl Anaconda's P

a c k a g e M a n a g e r c o n d a

The Anaconda developers built conda to address

the prob-
lems they've seen with `pip` and other tools. `pip` is a Python pack-
age man-
ager, but `conda` works with any soft-
ware and lan-
guage. `c`

`o`
`n`
`d`
`a`
also
avoids
the
need
for
some-
thing
like
`v`
`i`
`r`
`t`
`u`
`a`
`l`
`e`
`n`
`v`
to
keep
in-
stal-
la-
tions
from
step-
ping
on
one
an-
other.

If

you
in-
stalled
the
Anaconda
dis-
tri-
bu-
tion,
you
al-
ready
have
the
c
o
n
d
a
pro-
gram.
If
not,
you
can
get
Python
3
and
c
o
n
d
a
from
the
mini-

conda page. As with Anaconda, make sure the file you down-load starts with `Miniconda3`; if it starts with `Miniconda`

alone, it's the Python 2 version.

`conda` works with `pip`. Although it has its own public package repository, commands like `conda`

`asearch` will also search the PyPi repository. If you have problems with `pip`, `conda` might be a good alternative.