

Setting up your Development Environment

Alex Flückiger

Spring 2020

Contents

1	Introduction	2
2	Installation Guide for macOS	2
2.1	Install Development Environment Xcode	2
2.2	Install Package Manager Homebrew	3
2.3	Install Python 3 with Anaconda Platform	3
2.4	Install command-line Tools	5
3	Installation Guide for Windows	5
3.1	Install Python 3 with Anaconda Platform	5
3.2	Install Ubuntu in a Windows Subsystem	6
3.3	Install various command-line tools	7
4	Installation Guide for Linux	7
5	First Steps in Python	7

1 Introduction

Depending on your operating system, you need to install various components to be able to program in Python and perform common data wrangling tasks in the command-line when preparing the textual data.

Please note that the installation and setup of software are sometimes harder and much more poorly documented than mere usage. Moreover, there are many different ways to achieve this. Thus, this guide aims to ease the proper set up of the development environment for Windows 10 and macOS. Specifically, the proposed installation reconciles a cross-platform usage of the tools with relative simplicity (e.g., no virtual environments). Before you proceed with the installation, make sure that you run a current system version, that it has at least 15 GB free space, and you have a backup of your files.

The easiest way to install Python 3 is via the Anaconda platform, which includes standard packages already and also comes with a handy application such as [Jupyter Notebook](#) and [Spyder](#). Spyder is a cross-platform integrated development environment (IDE) for the Python language similar to what RStudio is for the programming language R.

The guide provides instructions to install the following components:

- Python 3 with Anaconda Platform (programming language)
- Tesseract (Optical Character Recognition)
- various Bash tools

Please keep in mind that there is *no* feedback in the command-line unless there is an issue. At first, the missing confirmations of successful actions may be confusing. However, it just means the command was executed accordingly; thus, there is no need to bother you with further messages.

2 Installation Guide for macOS

2.1 Install Development Environment Xcode

Xcode is an integrated development environment (IDE) comprising various software development tools for macOS. Among other things, the installation of Xcode is a requirement for the subsequent installation of the package manager Homebrew. To check if you have Xcode already installed, you need the command-line.

1. Open a Terminal to get a command-line interface. When you cannot find the application in your system tray, press the **command** and **spacebar** keys to search and type **Terminal** to search for it.
2. To check if you have installed Xcode already, type your in the Terminal window:

```
xcode-select -p
```

3. If you receive the following output, then Xcode is installed:
`/Library/Developer/CommandLineTools`. If you see an error, then install [Xcode from the App Store](#) via your web browser and accept the default options.
4. Once Xcode is installed, return to your Terminal window. Next, you'll need to install Xcode's separate Command Line Tools app, which you can do by typing:

```
xcode-select --install
```

5. At this point, Xcode and its Command Line Tools app are fully installed, and we are ready to install the package manager Homebrew.

Source: [Digital Ocean](#)

2.2 Install Package Manager Homebrew

1. To install Homebrew, type the following command into your Terminal window.

```
/bin/bash -c "$(curl -fsSL \
https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

When there is an issue executing this command (i.e., unprintable characters), copy the identical looking installation command from the [official website](#) into your Terminal.

2. You can make sure that Homebrew was successfully installed by typing:

```
brew doctor
```

3. To ensure that your installation of Homebrew is up to date, run:

```
brew update
```

4. It is not needed for now, yet you can upgrade outdated packages altogether with:

```
brew upgrade
```

Source: [Homebrew](#)

2.3 Install Python 3 with Anaconda Platform

1. Download the 64-bit command-line installer for Python 3.7 for macOS from [here](#).
2. Open a Terminal window (command-line). The following command requires that the installer is located in the folder `Downloads` and has its original

name. If you have changed its location or name, then adapt the file path in the following command before execution:

```
bash ~/Downloads/Anaconda3-2019.10-MacOSX-x86_64.sh
```

3. In the subsequent installation dialogue, you can follow the suggestion by pressing **Enter** multiple times.

IMPORTANT: On the question `Do you wish the installer to initialize Anaconda3 by running conda init?`, you need to type in:

```
no
```

4. To activate the environment, you should enter in the Terminal window:

```
/Users/<Your username>/anaconda3/bin/activate
```

(Note: If you have forgotten your username, you can ask for it using `whoami` in the Terminal).

5. Then initialize the change with:

```
conda init zsh
```

6. After the initialization, you can close and reopen the terminal to activate the modification.

7. Finally, you can check the installation by entering `conda list` in the terminal. If you see a list of modules, you have successfully installed the Anaconda environment. You can start to program in Python.

8. To enable community-driven packages within Anaconda distribution, you also need to add a new channel:

```
conda config --append channels conda-forge
```

Source: [Towards Data Science](#), [conda-forge](#)

2.3.1 Install Python packages

In this step, you install a few non-standard Python packages with the Anaconda Package Manager `conda`. Due to an [issue](#) in the current version of Anaconda, it is required to install the packages in a virtual environment with the following commands in a Terminal:

```
conda update -n base -c defaults conda
conda create --name myenv
conda activate myenv
conda install spyder
```

Subsequently, you can run these commands in your Terminal window (command-line):

```
conda install -c conda-forge spacy
conda install -c conda-forge textacy
conda install -c conda-forge plotnine
```

2.4 Install command-line Tools

2.4.1 Zsh

macOS Catalina introduced Zsh as the new default shell instead of Bash. To enable Zsh as new default on your system, type:

```
chsh -s /bin/zsh
```

2.4.2 Tesseract

1. Install the text recognition engine Tesseract, which allows extracting text from images, with:

```
brew install tesseract
```

2. Install the various language models for Tesseract with:

```
brew install tesseract-lang
```

Source: [Tesseract](#)

2.4.3 wget

1. Install the tool wget that allows you to retrieve content from web servers via the command-line with:

```
brew install wget
```

3 Installation Guide for Windows

3.1 Install Python 3 with Anaconda Platform

1. We download the 64-bit graphical installer for Python 3.7 for Windows from [here](#).
2. After the download is complete, you can execute the downloaded file to start the installation process.
IMPORTANT: Under **Advanced Options** you should tick the checkbox **Add Anaconda to my PATH environment variable**. Otherwise you can accept the recommended options.
3. To enable community-driven packages within Anaconda distribution, you also need to add a new channel. Open a Anaconda Prompt from your application list and execute:

```
conda config --append channels conda-forge
```

Source: [Anaconda](#), [conda-forge](#)

3.1.1 Install Python packages

In this step, you install a few non-standard Python packages with the Anaconda Package Manager conda. Due to an [issue](#) in the current version of Anaconda, it is required to install the packages in a virtual environment with the following commands using the same Anaconda Prompt as before:

```
conda update -n base -c defaults conda
conda create --name myenv
conda activate myenv
conda install spyder
```

Subsequently, you can run these commands:

```
conda install -c conda-forge spacy
conda install -c conda-forge textacy
conda install -c conda-forge plotnine
```

3.2 Install Ubuntu in a Windows Subsystem

To use the powerful bash tools on Windows, you install a Linux Ubuntu system within your Windows environment.

1. Before installing any Linux distros for WSL, you must ensure that the “Windows Subsystem for Linux” optional feature is enabled. Open PowerShell as Administrator by right-clicking on the application icon and run the following command in the shell:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
```

2. Restart your computer when prompted.
3. Install Linux Ubuntu 18.04 LTS via the [Microsoft Store](#)
4. From the distro’s page, select **Get**.
5. To complete the initialization of your newly installed distro, launch a new instance. You can do this by clicking the “launch” button in the Microsoft Store app, or launching the distro from the Start menu.
6. The first time a newly installed distribution runs, a Console window will open, and you’ll be asked to wait for a minute or two for the installation to complete.
7. Once the installation is complete, you will be prompted to create a new user account (and its password).

IMPORTANT: Remember these credentials as they are used to switch to the administrator mode in your Linux system. You may choose the same account name and password as in your hosting Windows system.

8. Update your freshly installed Ubuntu system:

```
sudo apt update && sudo apt upgrade
```

9. Create symbolic links to your personal documents from your in the Ubuntu Bash for a easier access. For example, you may want to mount (i.e., make accessible) all the folders in your user directory in the home directory on Ubuntu as **personal**:

```
cd ~  
ln -s /mnt/c/Users/YOUR_WINDOWS_USERNAME/ personal/
```

10. Annoyingly, the copy/paste behavior is different in Windows Shells. Any selected Text is copied automatically and to paste it, you have to right-click on your mouse. You may want to reassign the shortcuts to **Ctrl+Shift+C** and **Ctrl+Shift+V** in the menu (**right-click** on the windows title bar → **Properties** → **Options**). Using **Ctrl+C** is not possible as it is used to cancel a running program.

Source: [Microsoft](#)

3.3 Install various command-line tools

- Open Ubuntu and install essential tools via the command-line:

```
sudo apt-get install build-essential
```

3.3.1 Tesseract

1. Install the text recognition engine Tesseract, which allows extracting text from images, with:

```
sudo apt install tesseract-ocr
```

2. Install the German language model for Tesseract with:

```
sudo apt install tesseract-ocr-deu
```

Source: [Tesseract](#)

4 Installation Guide for Linux

You presumably have all the necessary tools or you have the necessary knowledge to install them. Otherwise, you can ask me or just google for a manual. Anyway you are lucky as it is simpler than on the other platforms

5 First Steps in Python

Open Spyder, an editor for Python, that you have installed together with Anaconda. As a kind of initiation ritual, say hello to the world in Python by

issuing the following code lines:

```
print("Hello, World!")
```

Congrats, you wrote your first little program in Python. It may not be as impressive, but you can go along and will learn by practicing. The list of tutorials below provides a great starting point to learn the basics of Python by solving little exercises interactively.

- [Python Principles](#)
- [LearnPython](#)