

Beyond the Basics



- ❖ Where can I go from here?
- ❖ Other Python Libraries
- ❖ Third-Party Libraries

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

Welcome to the last week of class! There is so much to the Python world, and you may be wondering “Where do I go from here?” Python is used in so many ways, for many diverse applications. I’m going to talk about a few more Python standard libraries that are interesting and useful, and also introduce some third-party libraries for various things. I have placed a lot of links in the Resources file for this week. I can’t possibly cover everything in this one lesson – or even in another whole class!



❖ In Resources:

- ❖ Articles and blogs about improving code style
- ❖ Fluent Python by Luciano Ramalho - Great book
- ❖ Videos - mostly from past PyCon conventions
 - ❖ Watch my linked videos by Raymond Hettinger

I have tried to address being "pythonic" in the lectures as the occasion arose, so I'm not going to do it here specifically. However, there are links in the resources on making code more pythonic. The book *Fluent Python*, by Luciano Ramalho is really good in that regard also, especially if you are coming to Python from another programming language like C or Java. If nothing else, please watch the videos linked that are by Raymond Hettinger. He is a good speaker and I have enjoyed his talks immensely.

Useful Python Libraries



- ❖ We've seen:
 - ❖ random, math, argparse, csv, collections, itertools, functools, unittest, decimal, time
- ❖ Other standard library modules:
 - ❖ datetime, sqlite3, re, timeit, turtle
 - ❖ urllib, json, io, socket, webbrowser, ssl, email, xml
- ❖ Many more...

We've seen a number of Python libraries already in our class. There are many other standard libraries that are very useful. Some of them are datetime, sqlite3, re which is a regular expression module, timeit, and turtle, a graphics module. Then there are communication-related modules like urllib, json, io, socket, webbrowser, ssl, email, xml and many more.

Managing Third-Party Libraries



- ❖ Managing third-party libraries
 - ❖ PyPI - Python Package Index
 - ❖ Install with `'pip install package'`
 - ❖ Windows might need:
 - ❖ `'py -3 -m pip install package'`
 - ❖ Optional version-specific install:
 - ❖ `'pip install package==version_num'`
 - ❖ Upgrade a package:
 - ❖ `'pip install --upgrade package'`

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

4

There are many third-party packages that are very useful for programming in Python. The package installation program, `pip`, is used to install and manage Python packages.

Please remember that best practices for Python is that you always use a virtual environment. You don't want to install most third-party packages system-wide. So before installing third-party packages, be sure to have your virtual environment activated.

The Python Package Index (PyPI) is a central repository that contains third-party Python packages. Every package in the Python Package Index can be installed using the package manager `pip`, even though you wouldn't know it from looking at most of the entries in PyPI. This is something that can be confusing to beginners.

Oh, and I want to say that PyPI is not to be confused with PyPy, spelled P-Y-P-Y, which is an alternative implementation of the Python interpreter.

`Pip` is included in Python 2.7.9+ and 3.4+. If you are working with older versions of Python, you would need to do some extra work to install `setuptools` and `pip`, which we won't be covering here. You use `pip` to install packages, or to upgrade an installed package to the current version, add the `--upgrade` to the command

Using pip



- ❖ Using pip to install

- ❖ Example: install Django:

- ❖ `'pip install django'`

- ❖ Optional version-specific install:

- ❖ `'pip install django==1.9'`

- ❖ Uninstall

- ❖ `'pip uninstall django'`

- ❖ `'pip list'` displays installed packages

As an example, to install Django, a Python framework for web apps, you just type `'pip install django'`. Sometimes a package will have dependencies – if this is the case, pip will also install all the dependent packages. At the time of this writing, the current version of Django is 1.10. If you were working on a project that needed Django 1.9, then you could specify the version using double equal signs. You can also use pip to uninstall a package.

Pip list will list all the packages that are installed.

Using requirements.txt



- ❖ Save installed packages information:

- ❖ `'pip freeze > requirements.txt'`

- ❖ Install in new project:

- ❖ `'pip install -r requirements.txt'`

When you have a project that you might want to share – or maybe you want to move it to another computer – you need to be able to set up the project environment correctly and get the right packages installed. If someone else is going to work on the project, you want them to be able to set up their environment to match your project, otherwise you may have serious problems due to the different environments. The command “pip freeze” will output the package information in a format that pip can use. If that information is saved in a file “requirements.txt”, then in the new project, the same packages and package versions can be installed using the requirements.txt file with the pip command “pip install -r requirements.txt”.

Useful libraries and packages



- ❖ Many libraries and packages
 - ❖ Date/time
 - ❖ Scientific computing
 - ❖ Web & communication
 - ❖ Command line interfaces
 - ❖ Text analysis
 - ❖ Graphics
 - ❖ Other

I have some experience with a number of third party libraries, but I have to be honest and say that most of them I have not used, as they are outside my area of expertise. I will briefly go over packages for these subjects.

Dates/Times Libraries



- ❖ `datetime` - a standard library for handling dates & times
- ❖ `python-dateutil` - Provides more functionality
 - ❖ Relative dates/times
 - ❖ Special calculations such as Easter
- ❖ `pytz` - timezone-aware package
 - ❖ Because governments keep changing times
 - ❖ Accurate timezone conversion
- ❖ `dateparser`, `delorean`, `pendulum`, `arrow`

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

8

Let's talk about some libraries and third-party packages for dealing with dates and times. You might not realize how complicated dates and times are in real life. There are so many time zones - way more than the 24 you might naturally expect! The `datetime` standard library in Python is useful for handling dates & times. `Python-dateutil` provides more functionality, especially for working with relative dates/times like months and years that do not have the same number of days. It also has features for special calculations like Easter.

`Pytz` is a timezone-aware package. It keeps track of world timezones for accurate timezone conversions. This information is not in standard Python because governments keep changing timezones, changing whether or not they have daylight savings time and/or changing the dates of daylight savings time. So it can be updated whenever needed and can always be as correct as is possible. If they tried to keep this info in regular Python, you would have to update your Python to get correct time zones.

I've listed a few other specialized date/time packages that I have heard of: `Dateparser`, `Delorean`, `Pendulum`, and `Arrow`.

Scientific Computing Libraries



- ❖ SciPy - a Python-based ecosystem for mathematics, science, and engineering.
 - ❖ NumPy - n-dimensional array processing
 - ❖ Matplotlib - comprehensive 2-D plotting
 - ❖ Pandas - advanced data structures & analysis
 - ❖ SciPy library - fast scientific computing

- ❖ Scikit-learn - for machine learning

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

9

SciPy is a Python-based ecosystem of open-source software for mathematics, science, and engineering. Individual packages include NumPy, matplotlib, pandas, and a scipy library. Scikit-learn is a related system focused on machine learning.

Scientific Computing Frameworks



- ❖ Jupyter Notebook

- ❖ Access to SciPy libraries

- ❖ Anaconda

- ❖ Data Science Platform
 - ❖ Contains Jupyter notebooks and other tools

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

10

These frameworks are not Python packages although they use the Python libraries. Jupyter Notebook is a web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, machine learning and much more.

Anaconda Distribution gives superpowers to people that change the world with high performance, cross-platform Python and R that includes the best innovative data science from open source. Has over 720 packages for data preparation, data analysis, data visualization, machine learning and interactive data science applications.

Web-related packages



- ❖ Web frameworks:
 - ❖ Django - a full-featured web framework
 - ❖ Flask - a web micro-framework
 - ❖ Pyramid - an in-between web framework
- ❖ Web scraping:
 - ❖ BeautifulSoup - library for screen-scraping
 - ❖ Scrapy - for extracting data from websites
- ❖ Http Packages:
 - ❖ requests - “HTTP for Humans”
 - ❖ simplejson - simple, extensible JSON encoder and decoder
 - ❖ ujson - ultra fast JSON encoder and decoder

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

11

Web-related packages are very popular. For web frameworks, the primary ones are Django, Flask and Pyramid. Django is a full-featured framework for web applications. It is probably the most popular of the Python web frameworks and has many plug-ins and extensions to go with it.

Flask is a micro-framework for quick creation of simple web apps. It's a little easier to get started than Django, and is good for smaller applications. Pyramid is an in-between framework that advertises is easy to start but also easy to scale to a larger website.

For web scraping and analysis, I think the most popular packages are BeautifulSoup and Scrapy - both of them are great for getting website data and manipulating it for analysis.

Other web communications-related packages are requests, advertised as "http for humans", simplejson, a simple and extensible JSON encoder and decoder, and uJson, billed as an ultrafast JSON encoder and decoder. JSON is a format for encoding data that actually looks very much like Python dictionaries.

Command Line Interfaces



- ❖ We saw std library argparse in Week 2
- ❖ getopt - std library simple argument parser
- ❖ docopt - define command-line argument rules with usage string
- ❖ Clint and click - newer tools for command line applications

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

12

There are a number of packages that are useful for command line programs. We saw argparse already in the lesson from Week 2. Getopt is in the standard library and is a traditional argument parser with very few bells and whistles. It is written in C and works similarly to the C getopt() function. I suspect some former C programmers got together to make this library. But I prefer argparse or docopt over getopt; I feel that getopt is unnecessarily complicated, especially with the ease of use of the other packages.

With the package docopt, we provide a usage string and it infers the option rules on its own. It's sort of the reverse of what happens with argparse, where argparse.ArgumentParser accepts option rules and generates the usage string for us. Then there is clint, which is a set of tools for developing command line programs. It includes things like color output and progress bars. It is still in beta, I believe, but looks promising. I have also heard good things about the package Click, which is also very new.

Text Analysis



- ❖ Textblob - processing textual data
- ❖ Textmining - statistical text mining
- ❖ Fuzzy - using phonetic algorithms
- ❖ Whoosh - text indexing and search

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

13

Textblob is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Textmining package contains a variety of useful functions for text mining in Python. It focuses on statistical text mining (i.e. the bag-of-words model) and makes it very easy to create a term-document matrix from a collection of documents. This matrix can then be read into a statistical package (R, MATLAB, etc.) for further analysis.

Fuzzy is a python library implementing common phonetic algorithms quickly.

Whoosh is a fast, featureful full-text indexing and searching library implemented in pure Python. Programmers can use it to easily add search functionality to their applications and websites.



- ❖ Pyro-Robotics - robotics environment for teaching and research
- ❖ RobotPy - for FIRST robotics competitions
- ❖ Gopigo - for GoPiGo Raspberry Pi robot kits

There are a number of Python libraries for robotics. These are a few of the ones I've heard of recently that people are actually using. Pyro Robotics is a set of objects, environment, and libraries for exploring robotics. Can be used to do very simple control (like `robot.move()`), or more complex planning and learning problems.

RobotPy is a community of FIRST mentors and students dedicated to developing python-related projects for the FIRST Robotics Competition. There are a number of robotpy-related packages in PyPI.

GoPiGo – is for use with the raspberry Pi robot kits. They are made by Dexter industries and are completely open source.



- ❖ turtle - good teaching library
- ❖ tkinter - interface to Tcl/Tk for graphical user interfaces
- ❖ Kivy - Support for multi-touch on all platforms and phones too.
- ❖ Pygame - based on SDL

Turtle graphics is a popular way for introducing programming to kids. Visualize a turtle whose tail has a pen attached to it. Commands are simple like pen down to draw, pen up, forward, backward, left, right, etc. The pen that the turtle has makes the graphics.

Tkinter stands for “Tk interface”, and is the standard Python interface to the Tk GUI toolkit. Tk is available on most unix-based platforms and on Windows. It provides basic tools for making a GUI application, with buttons and graphics, etc. provided.

Kivy is an Open source Python library for rapid development of applications that make use of innovative user interfaces, such as multi-touch apps. Kivy runs on Linux, Windows, OS X, Android and iOS. You can run the same code on all supported platforms.

PyGame is a Python wrapper for the SDL multimedia library. SDL stands for Simple DirectMedia Layer, which is a cross-platform development library designed to provide low level access to audio, keyboard, mouse, joystick, and graphics hardware via OpenGL and Direct3D.

Other packages



- ❖ Flake8 - you should use flake8
- ❖ Xml, lxml - for processing xml files
- ❖ Defusedxml - for safe xml processing
- ❖ Re - Regular expression built-in
- ❖ Regex - a replacement package

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

16

Here are a few other useful packages. Flake8 is a combination code style checker, linter and McCabe code complexity analyzer. Once it's installed, you can just type flake8 on the command line and it will look at all the files in that folder. Note that if your virtual environment folder is in the same folder with your files, don't do this! If this is the case, always list the files for flake8 to look at. Otherwise you will be overwhelmed with messages from all the python code in the virtual environment folder. Much of it doesn't pass flake8, some from legacy code and some due to changes for efficiency's sake. I highly recommend that you always use flake8 when developing software.

The xml parser works great, but there is a third-party library called lxml which parses much faster than the built-in xml module. The lxml library follows the same parsing rules as the xml module, so it can be used as a drop-in replacement for the xml module. Both the xml and lxml modules are vulnerable to a number of XML parser attacks. The defusedxml library is a wrapper around Python XML libraries that conform to the standard xml parsing. You can use defusedxml with either the xml module or the lxml library.

Re is a standard library module for working with regular expressions, and regex is a third-party replacement. Regular expressions are a way to specify search patterns. It is something everyone should learn a little bit, but it can be a huge subject by itself. There is a lot to regular expressions; they are very powerful, but are cryptic and easy to get complicated and confusing.

Other Packages, cont.



- ❖ Sqlite3 - for SQLite databases
- ❖ Timeit - code timer

Sqlite3 is a standard library package that provides a lightweight disk-based database that doesn't require a separate server process. It's fairly simple for a database and makes it possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

The timeit module provides a simple way to time small bits of Python code. It has both a Command-Line Interface as well as a callable one. It's very useful to check little pieces of code to find out where a bottleneck might be occurring.

Stay Connected!



- ❖ [Pyvideo.org](https://pyvideo.org) - Python videos from events
- ❖ Meetups - look for local groups
- ❖ Open source projects

© Diane Chen 2016-2017 UCSD Extension Online Learning Course: CSE-41273 Python Programming Fundamentals

18

I hope you will continue learning Python and being productive with it. There are many ways of finding out more information about Python. One good way is to check out the website pyvideo.org. They have cataloged a huge number of Python videos from Python conventions worldwide, including conferences for django, data science and others.

Check [meetup.com](https://www.meetup.com) for python meetup groups near you. I am very active with the San Diego Python meetup and PyLadies groups.

Something you might want to consider is contributing to open source projects. Many of them are very beginner-friendly and even have some of the issues tagged for beginners. If you have the opportunity to go to any Python conventions, I urge you to stay for the sprint days. These get-togethers are great, because you will meet people who are experienced with projects and are willing to help beginners get started on the projects and will provide guidance on the process. Having contributions to open source projects in your GitHub profile can help you get a job, because people can see your work and they know you have experience working with others.