Introduction to Python and PyMARC

Session I: Installation and Basic Syntax

Lauren Magnuson, California State University, San Marcos

Hosted by ALCTS: Association for Library Collections & Technical Services
Session I Overview

- Why Python? Why not MARCEdit?
- Resources
- Python Versions
- Installing Python and the PyMARC Library
  - Mac
  - Windows
- Use Python on Command Line
- Using PyMARC scripts (GitHub Repository)
About Me

- Currently Head of Collection Management and Technical Services, California State University San Marcos (North of San Diego) and Development Coordinator for the PALNI Library Consortium in Indiana
- Started out by taking programming classes in now basically obsolete languages (ASP.NET, ActionScript (ugh))
- Prefer PHP and Python
- GitHub: https://github.com/lpmagnuson
Why Python?

Python is great for writing scripts to:

- Clean up
- Transform
- Analyze

Data

Including:
- CSV (csv module)
- MARC (PyMARC module - the focus of this presentation)
- XML (ElementTree module)
- Linked Data: RDF (RDFLib), JSONLD (rdflib-jsonld), SPARQL (sparqlwrapper)
Why Not MARCEdit?

**MARCEdit** is awesome. Python/PyMARC does not replace MARCEdit. Python scripts can supplement your MARCEdit workflows.

If you find you **frequently** need to process data in **BOTH** MARCEdit AND Excel, that process is a good candidate for creating a Python/PyMARC workflow.
Sample PyMARC Projects

Transform a file of MARC records for bulk DSpace metadata ingest

Extract and transform ISBNs for holdings upload for GOBI

Generate a KBART file from MARC records for custom WorldCat Knowledge Base collections

Extract and transform OCLC numbers from a set of MARC records
Version 2 or 3?

There are two versions of Python - Python 2 and Python 3

- Some functions are not compatible between versions (Python 3 is not fully backward compatible with Python 2)
- Best to use Python 3; Python 2 is nearing end of support (April 12, 2020)
- Some older scripts may not work in Python 3
- Examples discussed in these sessions presume Python 3
Resources

Mac OSX instructions:

http://docs.python-guide.org/en/latest/starting/install/osx/

PC instructions:

http://docs.python-guide.org/en/latest/starting/install/win/

PyMARC instructions:

https://github.com/edsu/pymarc
Resources, Continued

PyMARC Google Group:

https://groups.google.com/forum/#!forum/pymarc

Stack Overflow (Programming Forum):

https://stackoverflow.com

When in doubt, Google it! (seriously, just copy and paste your error into Google)
Recommended Text Editors

MAC/OSX:

- BBEdit

Windows/PC:

- Notepad++

There are fancier Integrated Development Environments (IDE’s) available, but I personally don’t use them.
Useful Unix Commands (for Terminal/PowerShell)

- `pwd` - outputs your current location on a computer/server
- `ls` - lists contents of the directory you’re in, e.g.,

```
ls /Users/username/Desktop/pymarc-workshop
```

- `cd` - changes directories, e.g.,

```
cd /Users/username/Desktop/pymarc-workshop
```
Installing Python on Mac

- Open up Terminal
- `$ python3 --version`
  - If output looks like Python 3.6.2 (exact version may be different) you’re all set
- If Python 3 not installed, type `python --version`
  - Output will probably show Python 2.* (e.g., 2.7)
  - You can run some scripts with Python 2, but should install Python 3
Installing Python on Mac: Homebrew + PATH

1. Open Terminal
2. Install Xcode Command Line Tools: $ xcode-select --install
3. Install Homebrew
   
   ```bash
   $ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
   ```
4. Use Terminal to edit your PATH
   
   ```bash
   $ export PATH=/usr/local/bin:/usr/local/sbin:$PATH
   ```

Ctrl+X to Exit and Y to Save, then refresh your Terminal session:

```bash
$ source ~/.bash_profile
```
Installing Python on a Mac: Install Python3

4. Next, in Terminal, use Homebrew to install Python 3:

   $ brew install Python3

5. After installation completes, test installation with following Terminal command:

   $ python3 --version

   Output should be:   Python  3.6.2
Installing Python on a Mac: PyMARC

6. Make sure Pip is installed (Pip is a package manager for Python):

   $ pip3 -V

   Output should be:

   pip 9.0.1 from /usr/local/lib/python3.6/site-packages (python 3.6)

7. Use Pip to install the PyMARC library:

   $ pip3 install pymarc
Install Python on Windows, Step 1: Download

Visit the Python for Windows download page, and select the executable installer (likely the 64 bit version)

- **Python 3.6.1 - 2017-03-21**
  - Download Windows x86 web-based installer
  - Download Windows x86 executable installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86-64 web-based installer
  - **Download Windows x86-64 executable installer**
  - Download Windows x86-64 embeddable zip file
  - Download Windows help file

Image Source: https://www.howtogeek.com/197947/how-to-install-python-on-windows/
Install Python for Windows, Step 2: Install

On the first screen, enable the “Add Python 3.x to PATH” option and then click “Install Now.”

If prompted, select “Disable path length limit”
Install Python For Windows, Step 3: Editing Path

To test that Python is installed, open up your Windows PowerShell (comes with Windows 7 and above), and type `python --version`. If you get an error (command not found), you’ll need to add an ‘environment variable’ to ensure that Windows knows where to look for the Python library.

PowerShell can be found by using the search feature of the Windows start menu to search for PowerShell.
Install Python For Windows, Step 3: Editing Path

Hit Start, type “advanced system settings,” and then select the “View advanced system settings” option. In the “System Properties” window that opens, on the “Advanced” tab, click the “Environment Variables” button.

Image Source: https://www.howtogeek.com/197947/how-to-install-python-on-windows/
Install Python For Windows, Step 3: Editing Path

Next, in the “System Variables” section, find the Path variable and click Edit....

Scroll over to the end of the value (do not delete anything) and add this (be sure to change ‘username’ to your login username:

;C:\Users\username\AppData\Local\Programs\Python\Python 36

Close and re-open PowerShell, and try python --version again. Your output should look like Python 3.6.2.

Image Source: https://www.howtogeek.com/197947/how-to-install-python-on-windows/
Troubleshooting

If you're still not getting Powershell to recognize Python when you type python --version, try the following:

- Verify where Python actually installed. By default on Windows, it should install under `\Users\username\AppData\Local\Programs\Python\Python36` if using Python 3.6.*, but could have installed somewhere else. Try searching your computer for Python, and verify the version you installed (e.g., if using Python 3.7, look for Python37)
- Try using python3 --version instead of python --version at the command line
- Google the error you are seeing, or contact me
Install Python for Windows, Step 4: PyMARC

In the command line type:

```
pip3 install pymarc
```

If you get an error about pip not being found, download pip: [https://bootstrap.pypa.io/get-pip.py](https://bootstrap.pypa.io/get-pip.py). Right-click anywhere on this page to download the pip file to your computer. Use PowerShell to navigate to the directory in which you saved the file, and run:

```
python get-pip.py
```

Or

```
python3 get-pip.py
```
Use Python on the Command Line

You can run Python code on the Command Line by entering python or python3 in your command line tool (Terminal or PowerShell):

```
python
```

Python 3.6.2 (default, Jul 17 2017, 16:44:47)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)] on darwin
Type "help", "copyright", "credits" or "license" for more information.

```
Use Python on the Command Line

Print something:

```python
>>> print("hello world")
hello world
```  

Split a string (text phrase) - get the last 4 characters of a string:

```python
>>> mystr = "Room 1020"
>>> mystr[-4:]
'1020'
```
GitHub Script Package

https://github.com/lpmagnuson/pymarc-workshop.git

This code repository contains a set of MARC records and PyMARC sample scripts designed to extract OCLC numbers, transform them, and then write extracted and transformed data elements to a CSV file.

Click “Clone or Download” and Download .zip file to a directory on your computer.

Use Terminal or PowerShell to navigate to the directory you downloaded to.
GitHub Script Package

Contents:

- marc.mrc / marc.mrk - a set of MARC records you can open with MARCEdit
- oclc1.py - outputs raw OCLC numbers (MARC 035|a field)
- oclc2.py - cleans up OCLC numbers
- oclc-regex.py - cleans up OCLC numbers with a Regular Expression
- oclc-regex-csv.py - Outputs cleaned up OCLC numbers to a CSV file for Excel
- isbn-gobi.py - (???) covered in next session :)

isbn-gobi.py - (???) covered in next session :)

isbn-gobi.py - (???) covered in next session :)
A Quick Look at marc.mrc (.mrk file included)
How to Run a Python Script

- Use your command line tool to navigate to the directory in which you’ve downloaded the files
- Typical syntax: python3 <script-name.py> or python <script-name.py>
  - e.g., python3 oclc1.py
- Note that python scripts are very space/position dependent. If your spacing is not correct, you may see this error (or similar):

  IndentationError: unindent does not match any outer indentation level

Avoid using the tab key when writing Python scripts! Use spacebar for consistent spacing within scripts. Check to make sure your conditionals (if/then statements) are properly aligned. This is one reason why many Python programmers prefer fancy IDE’s.
Yet More Resources

Regular Expressions (w/examples):

https://pymotw.com/3/re/index.html

Python LC Call Number Normalization (sorting call numbers the way they were meant to be sorted):

https://github.com/libraryhackers/library-callnumber-lc/tree/master/python
Python Tutorials

https://www.learnpython.org/

https://developers.google.com/edu/python/
Session II (Next Week)

● Detailed walkthrough of GitHub code examples
● How to create MARC records from non-MARC metadata using PyMARC
● More troubleshooting (encoding issues, etc.)
Questions?

Contact me:

Email: lmagnuson@csusm.edu

Twitter: @lpmagnuson