Python Standard Library List Documentation

Release 0.4.0

Jack Maney

February 23, 2017

Contents

This package includes lists of all of the standard libraries for Python 2.6, 2.7, 3.2, 3.3, and 3.4, along with the code for scraping the official Python docs to get said lists.

CHAPTER 1

Listing the modules in the standard library? Wait, why on Earth would you care about that?!

Because knowing whether or not a module is part of the standard library will come in handy in a project of mine. And I'm not the only one who would find this useful. Or, the TL;DR answer is that it's handy in situations when you're analyzing Python code and would like to find module dependencies.

After googling for a way to generate a list of Python standard libraries (and looking through the answers to the previously-linked Stack Overflow question), I decided that I didn't like the existing solutions. So, I started by writing a scraper for the TOC of the Python Module Index for each of the versions of Python above.

However, web scraping can be a fragile affair. Thanks to a suggestion by @ncoghlan, and some further help from @birkenfeld and @epc, the population of the lists is now done by grabbing and parsing the Sphinx object inventory for the official Python docs of each relevant version.

CHAPTER 2

Contents

Installation

The easy way is via `pip`:

```
pip install stdlib-list
```

The hard way is to clone this repository, go into the directory into which you cloned this repo, and do a

```
python setup.py install
```

Usage (Or: How To Get The List of Libraries)

The primary function that you'll care about in this package is `stdlib_list.stdlib_list`.

In particular:

```
In [1]: from stdlib_list import stdlib_list

In [2]: libs = stdlib_list("3.4")

In [3]: libs[:6]
Out[3]: ['__future__', '__main__', '_dummy_thread', '_thread', 'abc', 'aifc']
```

`stdlib_list.`**`stdlib_list`**(*version=None*)

Given a `version`, return a `list` of names of the Python Standard Libraries for that version. These names are obtained from the Sphinx inventory file (used in `sphinx.ext.intersphinx`).

Parameters **`version`** (*`str`*|*`None`*) – The version (as a string) whose list of libraries you want

(one of `"2.6"`, `"2.7"`, `"3.2"`, `"3.3"`, `"3.4"`, or `"3.5"`). If not specified, the current version of Python will be used.

Returns A list of standard libraries from the specified version of Python

Return type list

Fetching The Module Lists

The lists of modules themselves are grabbed from the Sphinx object inventory (ie the file used by `sphinx.ext.intersphinx` in order to build links to existing Python modules/functions/etc). You probably

shouldn't need to mess around with it. But if you want to, here you go.

stdlib_list.fetch.**fetch_list**(*version=None*)

For the given version of Python (or all versions if no version is set), this function:

•Uses the *fetch_inventory* function of :py:mod'sphinx.ext.intersphinx' to

grab and parse the Sphinx object inventory (ie `http://docs.python.org/<version>/objects.inv`) for the given version.

•Grabs the names of all of the modules in the parsed inventory data.

•Writes the sorted list of module names to file (within the *lists* subfolder).

Parameters **version** (*str|None*) – A specified version of Python. If not specified, then all

available versions of Python will have their inventory objects fetched and parsed, and have their module names written to file. (one of `"2.6"`, `"2.7"`, `"3.2"`, `"3.3"`, `"3.4"`, `"3.5"`, or `None`)

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

S

F

S