

PEP 206 -- Python Advanced Library

PEP:	206
Title:	Python Advanced Library
Author:	A.M. Kuchling <amk at amk.ca>
Status:	Withdrawn
Type:	Informational
Created:	
Post-History:	

Contents

- [Introduction](#)
- [Batteries Included Philosophy](#)
- [Domain: Web tasks](#)

Tweets by @ThePSF


Python Software Foundation 
 @ThePSF

Dig into Python Developers Survey 2019 raw data and share your findings! jb.gg/pythondevsurvey... [#pythondevsurvey](#)



Python Developers Survey 2019 Results
 Official Python Developers Survey 2019 by Python Soft...
jetbrains.com






Python Software Foundation 
 @ThePSF

Another [#ThrowbackThursday](#) pic from [#pycon2019](#) - awesome puzzle [@CapitalOneTech](#)! [@pycon](#)

[Embed](#)
[View on Twitter](#)

[Tweets by @ThePSF](#)

The PSF

The Python Software Foundation is the organization behind Python. Become a member of the PSF and help advance the software and our mission.

- [Domain: Scientific Programming](#)
- [Domain: Application Development](#)
- [Domain: Education](#)
- [Software covered by the GNU General Public License](#)
- [Open Issues](#)
- [Acknowledgements](#)

Introduction

This PEP describes the Python Advanced Library, a collection of high-quality and frequently-used third party extension modules.

Batteries Included Philosophy

The Python source distribution has long maintained the philosophy of "batteries included" -- having a rich and versatile standard library which is immediately available, without making the user download separate packages. This gives the Python language a head start in many projects.

However, the standard library modules aren't always the best choices for a job. Some library modules were quick hacks (e.g. `calendar`, `commands`), some were designed poorly and are now near-impossible to fix (`cgi`), and some have been rendered obsolete by other, more complete modules (`binascii` offers the same features as the `binhex`, `uu`, `base64` modules). This PEP describes a list of third-party modules that make Python more competitive for various application domains, forming the Python Advanced Library.

The deliverable is a set of scripts that will retrieve, build, and install the packages for a particular application domain. The Python Package Index now contains enough information to let software automatically find packages and download them, so the time is ripe to implement this.

Currently this document doesn't suggest *removing* modules from the standard library that are superseded by a third-party module. That's difficult to do because it entails many backward-compatibility problems, so it's not worth bothering with now.

Please suggest additional domains of interest.

Domain: Web tasks

XML parsing: ElementTree + SAX.

URL retrieval: libcurl? other possibilities?

HTML parsing: mxTidy? HTMLParser?

Async network I/O: Twisted

RDF parser: ???

HTTP serving: ???

HTTP cookie processing: ???

Web framework: A WSGI gateway, perhaps? Paste?

Graphics: PIL, Chaco.

Domain: Scientific Programming

Numeric: Numeric, SciPy

Graphics: PIL, Chaco.

Domain: Application Development

GUI toolkit: ???

Graphics: Reportlab for PDF generation.

Domain: Education

Graphics: PyGame

Software covered by the GNU General Public License

Some of these third-party modules are covered by the GNU General Public License and the GNU Lesser General Public License. Providing a script to download and install such packages, or even assembling all these packages into a single tarball or CD-ROM, shouldn't cause any difficulties with the GPL, under the "mere aggregation" clause of the license.

Open Issues

What other application domains are important?

Should this just be a set of Ubuntu or Debian packages? Compiling things such as PyGame can be very complicated and may be too difficult to automate.

Acknowledgements

The PEP is based on an earlier draft PEP by Moshe Zadka, titled "2.0 Batteries Included."

Source: <https://github.com/python/peps/blob/master/pep-0206.txt>

	About	Downloads	Documentation	Community	Success Stories	News
	Applications	All releases:	Docs	Community Survey	Arts	Python News
	Quotes	Source code	Audio/Visual Talks	Diversity	Business	PSF Newsletter
	Getting Started	Windows	Beginner's Guide	Mailing Lists	Education	Community News
	Help	Mac OS X	Developer's Guide	IRC	Engineering	PSF News
	Python Brochure	Other Platforms	FAQ	Forums	Government	PyCon News
	Events	License	Non-English Docs	PSF Annual Impact Report	Scientific	Contributing
	Python Events	Alternative Implementations	PEP Index	Python Conferences	Software Development	Developer's Guide
	User Group Events		Python Books	Special Interest Groups		Issue Tracker

Python Events Archive	Python Essays	Python Logo	python-dev list
User Group Events Archive		Python Wiki	Core Mentorship
Submit an Event		Merchandise	Report a Security Issue
	Community Awards		
Help & Contact	Diversity	Submit Website Bug	Code of Conduct Status ●
<small>Copyright ©2001-2020. Python Software Foundation Legal Statements Privacy Policy Powered by Heroku</small>			