Modules and Packages

Python's primary extension mechanism uses source code saved to modules and incorporated into a program through the import statement. The features that most developers think of as "Python" are actually implemented as the collection of modules called the Standard Library, the subject of this book. Although the import feature is built into the interpreter itself, there are several modules in the library related to the import process.

The importlib module exposes the underlying implementation of the import mechanism used by the interpreter. It can be used to import modules dynamically at runtime, instead of using the import statement to load them during start-up. Dynamically loading modules is useful when the name of a module that needs to be imported is not known in advance, such as for plugins or extensions to an application.

Python packages can include supporting resource files such as templates, default configuration files, images, and other data along with source code. The interface for accessing resource files in a portable way is implemented in the pkgutil module. It also includes support for modifying the import path for a package, so that the contents can be installed into multiple directories but appear as part of the same package.

zipimport provides a custom importer for modules and packages saved to ZIP archives. It is used to load Python EGG files, for example, and can also be used as a convenient way to package and distribute an application.

- importlib — Python's Import Mechanism
- pkgutil — Package Utilities
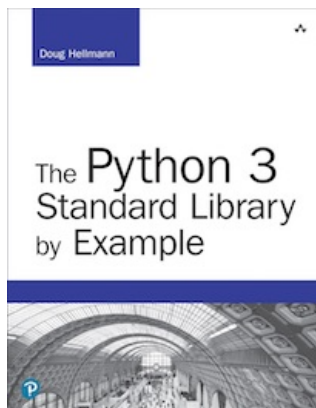- zipimport — Load Python Code from ZIP Archives

This page was last updated 2016-11-26.

Navigation

Get the book

The output from all the example programs from PyMOTW-3 has been generated with Python 3.7.1, unless otherwise noted. Some of the features described here may not be available in earlier versions of Python.

Looking for examples for Python 2?

Other Writing

✏️ Blog

📕 The Python Standard Library By Example