

Language Tools

In addition to the developer tools covered in an earlier chapter, Python also includes modules that provide access to its internal features. This chapter covers some of the tools for working in Python, regardless of the application area.

The [warnings](#) module is used to report non-fatal conditions or recoverable errors. A common example of a warning is the `DeprecationWarning` generated when a feature of the standard library has been superseded by a new class, interface, or module. Use warnings to report conditions that may need user attention, but are not fatal.

Defining a set of classes that conform to a common API can be a challenge when the API is defined by someone else or uses a lot of methods. A common way to work around this problem is to derive all of the new classes from a common base class, but it is not always obvious which methods should be overridden and which can fall back on the default behavior. Abstract base classes from the [abc](#) module formalize an API by explicitly marking the methods a class must provide in a way that prevents the class from being instantiated if it is not completely implemented. For example, many of Python's container types have abstract base classes defined in `abc` or [collections](#).

The [dis](#) module can be used to disassemble the byte-code version of a program to understand the steps the interpreter takes to run it. Looking at disassembled code can be useful when debugging performance or concurrency issues, since it exposes the atomic operations executed by the interpreter for each statement in a program.

The [inspect](#) module provides introspection support for all objects in the current process. That includes imported modules, class and function definitions, and the objects instantiated from them. Introspection can be used to generate documentation for source code, adapt behavior at runtime dynamically, or examine the execution environment for a program.

- [warnings](#) — Non-fatal Alerts
- [abc](#) — Abstract Base Classes
- [dis](#) — Python Bytecode Disassembler
- [inspect](#) — Inspect Live Objects

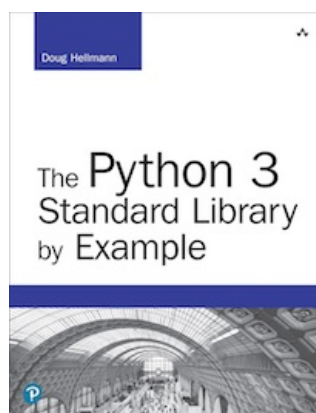
[← sysconfig — Interpreter Compile-time Configuration](#)

[warnings — Non-fatal Alerts →](#)

This page was last updated 2016-12-31.

Navigation

- [← sysconfig — Interpreter Compile-time Configuration](#)
- [warnings — Non-fatal Alerts](#)



[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?](#)

This Site

 [Module Index](#)

 [Index](#)



© Copyright 2019, Doug Hellmann



Other Writing

 [Blog](#)

 [The Python Standard Library By Example](#)