collections.abc — Abstract Base Classes for Containers

Purpose: Abstract base classes for container data types.

The `collections.abc` module contains abstract base classes that define the APIs for container data structures built into Python and provided by the `collections` module. Refer to the table below for a list of the classes and their purposes.

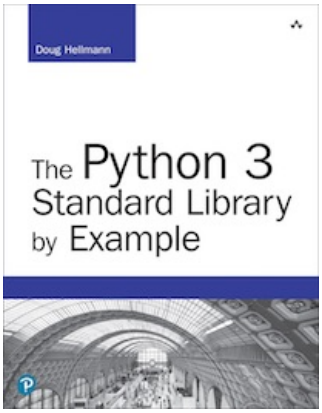
Abstract Base Classes

Class	Base Class(es)	API Purpose
Container		Basic container features, such as the `in` operator.
Hashable		Adds support for providing a hash value for the container instance.
Iterable		Can create an iterator over the container contents.
Iterator	Iterable	Is an iterator over the container contents.
Generator	Iterator	Extends iterators with the generator protocol from PEP 342.
Sized		Adds methods for containers that know how big they are.
Callable		For containers that can be invoked as a function.
Sequence	Sized, Iterable, Container	Supports retrieving individual items, iterating, and changing the order of items.
MutableSequence	Sequence	Supports adding and removing items to an instance after it has been created.
ByteString	Sequence	Combined API of `bytes` and `bytearray`.
Set	Sized, Iterable, Container	Supports set operations such as intersection and union.
MutableSet	Set	Adds methods for manipulating the set contents after it is created.
Mapping	Sized, Iterable, Container	Defines the read-only API used by `dict`.
MutableMapping	Mapping	Defines the methods for manipulating the contents of a mapping after it is created.
MappingView	Sized	Defines the view API for accessing a mapping from an iterator.
ItemsView	MappingView, Set	Part of the view API.
KeysView	MappingView, Set	Part of the view API.
ValuesView	MappingView	Part of the view API.
Awaitable		API for objects that can be used in `await` expressions, such as coroutines.
Coroutine	Awaitable	API for classes that implement the coroutine protocol.
AsyncIterable		API for iterables compatible with `async for`, as defined in PEP 492.
AsyncIterator	AsyncIterable	API for asynchronous iterators.

In addition to clearly defining the APIs for containers with different semantics, these abstract base classes can be used to test whether an object supports an API before invoking it using `isinstance()`. Some of the classes also provide implementations of methods, and they can be used as mix-ins to build up custom container types without implementing every method from scratch.

OrderedDict — Remember the Order Keys are Added to a Dictionary
array — Sequence of Fixed-type Data

This page was last updated 2017-01-28.



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?