Concrete Objects Layer

- Fundamental Objects
- Numeric Objects
- Sequence Objects
- Container Objects
- Function Objects
- Other Objects
- C API for extension modules

The functions in this chapter are specific to certain Python object types. Passing them an object of the wrong type is not a good idea; if you receive an object from a Python program and you are not sure that it has the right type, you must perform a type check first; for example, to check that an object is a dictionary, use `PyDict_Check`. The chapter is structured like the "family tree" of Python object types.

> [!WARNING] While the functions described in this chapter carefully check the type of the objects which are passed in, many of them do not check for `NULL` being passed instead of a valid object. Allowing `NULL` to be passed in can cause memory access violations and immediate termination of the interpreter.

Fundamental Objects

This section describes Python type objects and the singleton object `None`.

type.rst none.rst

Numeric Objects

pair: object; numeric

long.rst bool.rst float.rst complex.rst

Sequence Objects

pair: object; sequence

Generic operations on sequence objects were discussed in the previous chapter; this section deals with the specific kinds of sequence objects that are intrinsic to the Python language.

bytes.rst bytearray.rst unicode.rst tuple.rst list.rst

Container Objects

pair: object; mapping

dict.rst set.rst

Function Objects

function.rst method.rst cell.rst code.rst

Other Objects

file.rst module.rst iterator.rst descriptor.rst slice.rst memoryview.rst picklebuffer.rst weakref.rst capsule.rst sentinel.rst frame.rst gen.rst coro.rst contextvars.rst typehints.rst

C API for extension modules

curses.rst datetime.rst

□□□ #2
□ □□□ □□□ 2026-10-31 16:50:19 UTC
□ □□□ □□□ 2026-10-31 16:50:21 UTC