

具体的对象层

本章中的函数特定于某些 Python 对象类型。将错误类型的对象传递给它们并不是一个好主意；如果您从 Python 程序接收到一个对象，但不确定它是否具有正确的类型，则必须首先执行类型检查；例如，要检查对象是否为字典，请使用 [PyDict_Check\(\)](#)。本章的结构类似于 Python 对象类型的“家族树”。

警告： 虽然本章所描述的函数会仔细检查传入对象的类型，但是其中许多函数不会检查传入的对象是否为 `NULL`。允许传入 `NULL` 可能导致内存访问冲突和解释器的立即终止。

基本对象

本节描述 Python 类型对象和单一实例对象 `None`。

- [类型对象](#)
 - [创建堆分配类型](#)
- [None 对象](#)

数值对象

- [整数型对象](#)
 - [导出 API](#)
 - [PyLongWriter API](#)
- [布尔对象](#)
- [浮点数对象](#)
 - [打包与解包函数](#)
 - [打包函数](#)
 - [解包函数](#)
- [复数对象](#)
 - [表示复数的 C 结构体](#)
 - [表示复数的 Python 对象](#)

序列对象

序列对象的一般操作在前一章中讨论过；本节介绍 Python 语言固有的特定类型的序列对象。

- [bytes 对象](#)
- [字节数组对象](#)
 - [类型检查宏](#)
 - [直接 API 函数](#)
 - [宏](#)

- [Unicode对象和编解码器](#)
 - [Unicode对象](#)
 - [Unicode类型](#)
 - [Unicode字符属性](#)
 - [创建和访问 Unicode 字符串](#)
 - [语言区域编码格式](#)
 - [文件系统编码格式](#)
 - [wchar_t 支持](#)
 - [内置编解码器](#)
 - [泛型编解码器](#)
 - [UTF-8 编解码器](#)
 - [UTF-32 编解码器](#)
 - [UTF-16 编解码器](#)
 - [UTF-7 编解码器](#)
 - [Unicode-Escape 编解码器](#)
 - [Raw-Unicode-Escape 编解码器](#)
 - [Latin-1 编解码器](#)
 - [ASCII 编解码器](#)
 - [字符映射编解码器](#)
 - [Windows 中的 MBCS 编解码器](#)
 - [方法与槽位函数](#)
 - [PyUnicodeWriter](#)
 - [已弃用的 API](#)
- [元组对象](#)
- [结构序列对象](#)
- [列表对象](#)

容器对象

- [字典对象](#)
- [集合对象](#)

Function 对象

- [Function 对象](#)
- [实例方法对象](#)
- [方法对象](#)
- [Cell 对象](#)
- [代码对象](#)
- [代码对象标志](#)
- [附加信息](#)

其他对象

- 文件对象
- 模块对象
- 模块定义
 - 模块槽位
- 动态创建扩展模块
- 支持函数
 - 模块查找（单阶段初始化）
- 迭代器对象
- 描述符对象
- 切片对象
 - Ellipsis 对象
- MemoryView 对象
- 弱引用对象
- Capsule 对象
- 帧对象
 - 帧 locals 代理
 - 内部帧
- 生成器对象
- 协程对象
- 上下文变量对象
- DateTime 对象
- 类型注解对象