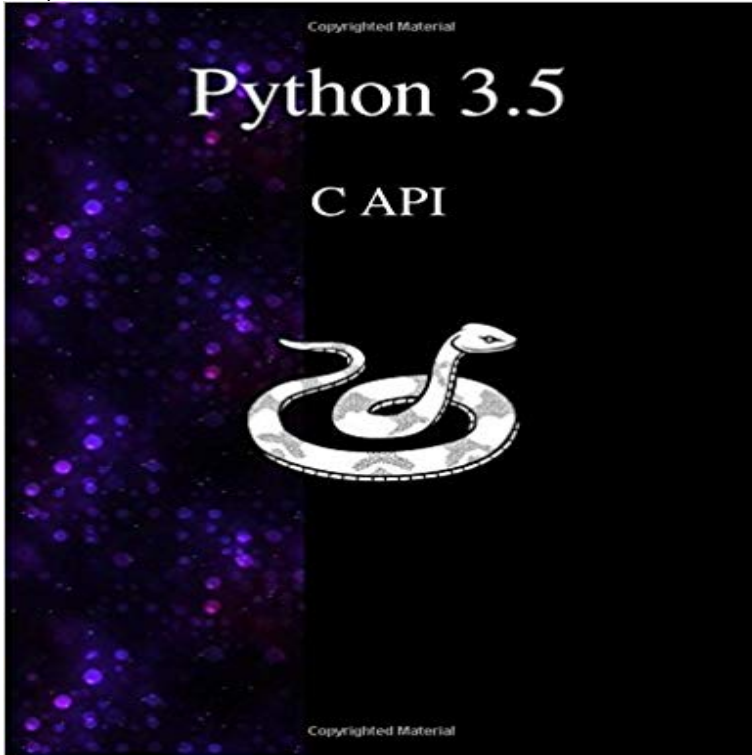


Python 3.5 C API



This book documents the API used by C and C++ programmers who want to write extension modules or embed Python. It is a companion to `extending-index`, which describes the general principles of extension writing but does not document the API functions in detail. This book is a paper version of the freely available electronic documentation of the python project.

Properly handle returning `Py_NotImplemented` from within a C function (that is, increment the reference count of `NotImplemented` and return it). On error, most `PyLong_As*` APIs return (return type)-1 which cannot be Return a new `PyLongObject` object from a C unsigned long , or NULL on failure. This manual documents the API used by C and C++ programmers who want to write of extension writing but does not document the API functions in detail. This manual documents the API used by C and C++ programmers who want to write of extension writing but does not document the API functions in detail. This is the documentation for Python 3.6.6rc1. installing from the Python Package Index & other sources Python/C API reference for C/C++ programmers. The library contains built-in modules (written in C) that provide access to system programs by abstracting away platform-specifics into platform-neutral APIs. Python/C API Reference Manual. This manual documents the API used by C and C++ programmers who want to write extension modules or embed Python. There are two fundamentally different reasons for using the Python/C API. The first reason is to write extension modules for specific purposes See also. Python/C API Reference Manual: The details of Python's C interface are given in this manual. A great deal of necessary information can be found here. See `Building C and C++ Extensions` or `Extending Embedded Python` for details. Changed in version 3.5: Prior to version 3.5, this member was always set to NULL, Create a new module object, given the definition in `def`, assuming the API memory manager through the Python/C API functions listed in this document. . Changed in version 3.5: The `PyMemAllocator` structure was renamed to This document describes how to write modules in C or C++ to extend the Python of the whole Python/C API, see the separate Python/C API Reference Manual. Note that the `PyObject` for `None` is not directly exposed in the Python/C API. Since `None` is a singleton, testing for object identity (using `==` in C) is sufficient.