1， pythoncpp.cpp 程序展示了 c++程序如何调用 python 的函数
（参考博客:[C++/Python] 如何在C++中使用一个Python类? (Use Python-defined class in C++)https://www.cnblogs.com/lancelod/p/4036922.html ）

2， 编译成功并运行 pythoncpp 程序时，在 libpython 文件夹下会自动生成 libpython.pyc，或者可以用 python 指令自己生成（python -m py_compile libpython.py）
（参考博客：Python 什么情况下会生成 pyc 文件？https://blog.csdn.net/m0_37550221/article/details/78907972 ）

3，python 中的数据类型

s (string) [char *]

Convert a null-terminated C string to a Python object. If the C string pointer is *NULL*, None is used.

s# (string) [char *, int]

Convert a C string and its length to a Python object. If the C string pointer is *NULL*, the length is ignored and None is returned.

z (string or None) [char *]

Same as s.

z# (string or None) [char *, int]

Same as s#.

u (Unicode string) [Py_UNICODE *]

Convert a null-terminated buffer of Unicode (UCS-2 or UCS-4) data to a Python Unicode object. If the Unicode buffer pointer is *NULL*, Noneis returned.

u# (Unicode string) [Py_UNICODE *, int]

Convert a Unicode (UCS-2 or UCS-4) data buffer and its length to a Python Unicode object. If the Unicode buffer pointer is *NULL*, the length is ignored and None is returned.

i (integer) [int]

Convert a plain C int to a Python integer object.

b (integer) [char]

Convert a plain C char to a Python integer object.

h (integer) [short int]

Convert a plain C short int to a Python integer object.

l (integer) [long int]

Convert a C long int to a Python integer object.

B (integer) [unsigned char]

Convert a C unsigned char to a Python integer object.

H (integer) [unsigned short int]

Convert a C unsigned short int to a Python integer object.

I (integer/long) [unsigned int]

Convert a C unsigned int to a Python integer object or a Python long integer object, if it is larger than sys.maxint.

k (integer/long) [unsigned long]

Convert a C unsigned long to a Python integer object or a Python long integer object, if it is larger than sys.maxint.

L (long) [PY_LONG_LONG]

Convert a C long long to a Python long integer object. Only available on platforms that support long long.

K (long) [unsigned PY_LONG_LONG]

Convert a C unsigned long long to a Python long integer object. Only available on platforms that support unsigned long long.

n (int) [Py_ssize_t]

Convert a C Py_ssize_t to a Python integer or long integer.

New in version 2.5.

c (string of length 1) [char]

Convert a C int representing a character to a Python string of length 1.

d (float) [double]

Convert a C double to a Python floating point number.

f (float) [float]

Same as d.

D (complex) [Py_complex *]

Convert a C Py_complex structure to a Python complex number.

O (object) [PyObject *]

Pass a Python object untouched (except for its reference count, which is incremented by one). If the object passed in is a *NULL* pointer, it is assumed that this was caused because the call producing the argument found an error and set an exception.

Therefore, Py_BuildValue()will return *NULL* but won't raise an exception. If no exception has been raised yet, SystemError is set.

S (object) [PyObject *]

Same as O.

N (object) [PyObject *]

Same as O, except it doesn't increment the reference count on the object. Useful when the object is created by a call to an object constructor in the argument list.

O& (object) [*converter*, *anything*]

Convert *anything* to a Python object through a *converter* function. The function is called with *anything* (which should be compatible withvoid *) as its argument and should return a "new" Python object, or *NULL* if an error occurred.

(items) (tuple) [*matching-items*]

Convert a sequence of C values to a Python tuple with the same number of items.

[items] (list) [*matching-items*]

Convert a sequence of C values to a Python list with the same number of items.

{items} (dictionary) [*matching-items*]

Convert a sequence of C values to a Python dictionary. Each pair of consecutive C values adds one item to the dictionary, serving as key and value, respectively.