Why does sys.getrefcount() return 2?

Asked 9 years, 11 months ago Modified 4 years, 6 months ago Viewed 8k times

22

As I understand, sys.getrefcount() returns the number of references of an object, which "should" be 1 in the following case:

```
import sys,numpy
a = numpy.array([1.2,3.4])
print sys.getrefcount(a)
```

3

However, it turned out to be 2! So, if I:

```
del a
```

Will the "numpy.array([1.2,3.4])" object still be there (no garbage collection)?

1 Answer

Sorted by: Highest score (default)

34

When you call `getrefcount()`, the reference is copied by value into the function's argument, temporarily bumping up the object's reference count. This is where the second reference comes from.

This is explained in the documentation:

> The count returned is generally one higher than you might expect, because it includes the (temporary) reference as an argument to `getrefcount()`.

As to your second question:

> If I "del a", will the "numpy.array([1.2,3.4])" object still be there (no garbage collection)?

By the time `getrefcount()` exits, the array's reference count will to back to 1, and a subsequent `del a` would release the memory.