Is it OK to use classes as dictionary keys in python?

I am about to write a function that finds a correct handler for a class. To achieve it I want to use a mapping from classes to handlers. I've already checked that it is possible to use classes as dictionary keys, however I am not completely sure if it is the right think to do.

My main concern is, if it is guaranteed that if I have a class imported in two different modules, these two will have the same hash. But maybe there are some other aspects that I should consider.

If idea of using classes itself is wrong, then I will use classnames, but that would require me to keep them unique, so I would prefer to use classes itself.

EDIT: I have made a test:

```
# file: a.py
import datetime
D = datetime.datetime

# file: b.py
import datetime
from a import D
print hash(D) == hash(datetime.datetime)
```

This prints "True", but I am still not sure if there is a way for it to be False for the same class.

python class dictionary hashmap

edited Feb 29 '12 at 9:39

asked Feb 29 '12 at 9:16
zefciu
1,202 10 28

4 Answers

It's perfectly valid, but note that what you are doing is fake-adding a method to that class.

Consider whether it would be better to just add appropriate methods to your classes, either in the normal way, or possibly just by guerilla-patching. (It might not be, but consider it).

answered Feb 29 '12 at 13:17
Marcin
32.5k 10 81 155

1 Thanks. I still believe that it is necessary for the system I'm trying to implement, but thanks for the warning. – zefciu Feb 29 '12 at 16:26

This is perfectly okay, since classes are objects themselves. For objects the hash key is their memory location which is guranteed to be unique. Let me get a reference...

answered Feb 29 '12 at 9:18
Constantinius
20.2k 6 43 74

I am not worried about lack of uniqueness (that would yield false positives). I am worried about the same class stored in two places in memory (which would yield false negative). – zefciu Feb 29 '12 at 9:35

2 @zefciu, The same class cannot be stored in two places in memory, otherwise it's not the same... – warvariuc Feb 29 '12 at 9:52

It's perfectly valid to use a class as a dictionary key, I've done it a couple of times already. However, inheritance will not work, that is, a handler for a class will not apply to its subclasses which may not be what you want. `__mro__` could be useful in that case, but this solution seems wrong to me.

answered Feb 29 '12 at 13:07

Sylvain Prat

159 1 10

The hash, unless you explicitly override it, will be the `id()` - which is the memory location.

So any two classes will have a different hash, unless it is two references to the same instance. If it is a class set up at import time of a module, then it *would* be the same instance, because the code at import time is executed once. In all other cases I can think of, it would be different instances.

answered Feb 29 '12 at 9:19

wim

108k 28 201 314