How do I most efficienty check the unique elements in a list?

Asked 14 years, 8 months ago Modified 14 years, 8 months ago Viewed 376 times

5

let's say I have a list

```
li = [{'q':'apple','code':'2B'},
      {'q':'orange','code':'2A'},
      {'q':'plum','code':'2A'}]
```

What is the most efficient way to return the count of unique "codes" in this list? In this case, the unique codes is 2, because only *2B* and *2A* are unique.

I could put everything in a list and compare, but is this really efficient?

python list dictionary performance

Share
Improve this question
Follow

edited Apr 2, 2010 at 22:44
Eli Courtwright
193k ● 68 ● 218 ● 257

asked Apr 2, 2010 at 22:43
TIMEX
271k ● 364 ● 798 ● 1.1k

2 Duplicate of all of these: stackoverflow.com/search?q=%5Bpython%5D+duplicate+list.
Specifically this: stackoverflow.com/questions/1143379/... – S.Lott Apr 3, 2010 at 2:40

1 Answer

Sorted by: Highest score (default)

8

Probably the most efficient simple way is to create a set of the codes, which will filter out uniques, then get the number of elements in that set:

```
count = len(set(d["code"] for d in li))
```

As always, I advise to not worry about this kind of efficiency unless you've measured your performance and seen that it's a problem. I usually think only about code clarity when writing this kind of code, and then come back and tighten it only if I've profiled and found that I **need** to make performance improvements.

Share

edited Apr 2, 2010 at 22:50

answered Apr 2, 2010 at 22:45

Improve this answer

Follow

Eli Courtwright

193k ● 68 ● 218 ● 257

1 FWIW when sets were first introduced in Python this was a novel concept to me. I found it very useful to read up on the math behind sets, which ties it all together: en.wikipedia.org/wiki/Set_(mathematics) – jathanism Apr 2, 2010 at 23:42

2 To take the unique of some *group*, it's always O(n), since you have look at every element, unless you have some other inside information. – Gregg Lind Apr 3, 2010 at 0:11