What is the difference between sets and lists in Python?

Is the only difference between sets and lists in Python the fact that you can use the union, intersect, difference, symmetric difference functions to compare two sets? Why can't these functions simply be applied to lists? In what situations are sets more useful than lists?

python list set

edited Sep 10 '12 at 15:11
DSM
147k 16 272 288

asked Sep 10 '12 at 15:09
user1588869
101 1 1 7

Sets, by definition, don't include duplicate values; I assume that Python, like other languages, will allow duplicate entries in a list. – Anthony Grist Sep 10 '12 at 15:11

1 This is laid out pretty well in the documentation. Sets and Lists – Scott Bartell Sep 10 '12 at 15:35

3 Answers

There's a huge difference.

1. Sets can't contain duplicates
2. Sets are unordered
3. In order to find an element in a set, a hash lookup is used (which is why sets are unordered). This makes `__contains__` (`in` operator) a lot more efficient for sets than lists.
4. Sets can only contain hashable items (see #3). If you try: `set(([1],[2]))` you'll get a `TypeError`.

In practical applications, lists are very nice to sort and have order while sets are nice to use when you don't want duplicates and don't care about order.

Also note that if you don't care about order, etc, you can use

```
new_set = myset.intersection(mylist)
```

to get the intersection between a `set` *and a* `list`.

edited Sep 10 '12 at 15:49

answered Sep 10 '12 at 15:11
mgilson
170k 26 289 398

Also, sets can contain only objects that are hashable; lists can contain any kind of object. – kindall Sep 10 '12 at 15:47

@kindall -- That's implicit in the explanation for #3, but I've added it as #4 to make it more explicit. Thanks for the comment. – mgilson Sep 10 '12 at 15:50

`sets` — Unordered collections of unique elements

`lists` - ordered collections of elements

`sets` allows you to do operations such as `intersection` , `union` , `difference` , and `symmetric difference` , i.e operations of math's set theory. Sets doesn't allow indexing and are implemented on hash tables.

`lists` are really variable-length arrays, not Lisp-style linked lists. In lists the elements are accessed by indices.

edited Jul 11 '16 at 11:13

answered Sep 10 '12 at 15:12

Ashwini Chaudhary

142k 23 230 304

Some more differences are:

1. List can be 2-D whereas a set can't.
2. As list are ordered (IE. have serial number) list are comparatively slow to execute whereas sets are fast.
3. List in python is like Array of java or c.
4. Printing a set almost always provide different sequence of output.

edited Mar 9 at 8:42

Usman Maqbool

1,507 5 11 29

answered Aug 31 '16 at 3:56

Niladri Sekhar Basu

53 8

1 a set can be of as many dimensions as you want as long as you use `frozenset` – Copperfield Aug 31 '16 at 4:28