

Join the Stack Overflow Community

Stack Overflow is a community of 6.8 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

In Python, when to use a Dictionary, List or Set?

When should I use a dictionary, list or set?

Are there scenarios that are more suited for each data type?

[python](#) [list](#) [dictionary](#) [set](#)

edited Nov 8 '15 at 3:29



[Student](#)

331 5 20

asked Aug 15 '10 at 20:22



[Blankman](#)

70.3k 210 581 955

5 Answers

A `list` keeps order, `dict` and `set` don't: when you care about order, therefore, you must use `list` (if your choice of containers is limited to these three, of course;-).

`dict` associates with each key a value, while `list` and `set` just contain values: very different use cases, obviously.

`set` requires items to be hashable, `list` doesn't: if you have non-hashable items, therefore, you cannot use `set` and must instead use `list`.

`set` forbids duplicates, `list` does not: also a crucial distinction. (A "multiset", which maps duplicates into a different count for items present more than once, can be found in `collections.Counter` -- you could build one as a `dict`, if for some weird reason you couldn't `import collections`, or, in pre-2.7 Python as a `collections.defaultdict(int)`, using the items as keys and the associated value as the count).

Checking for membership of a value in a `set` (or `dict`, for keys) is blazingly fast (taking about a constant, short time), while in a list it takes time proportional to the list's length in the average and worst cases. So, if you have hashable items, don't care either way about order or duplicates, and want speedy membership checking, `set` is better than `list`.

answered Aug 15 '10 at 20:30



[Alex Martelli](#)

504k 92 901 1179

35 I wish I could donate my answer votes to yours... Hey everyone, this is where the meat is :)

– [Jon Skeet](#) Aug 15 '10 at 20:44

I took the bait! thanks for the summary! – [Blankman](#) Aug 15 '10 at 20:49

- 6 @Jon, thanks! Hmmm, maybe the idea of "transferring upvotes" (at some "discount rate", e.g., 2 for 1) is worth discussing in meta...?) I know I've felt similarly on certain occasions... – [Alex Martelli](#) Aug 15 '10 at 21:20

I could use a list to build a list of unique items, like a set; so is the main reason to use a set speed? – [James](#) Nov 28 '16 at 3:28

- Do you just need an ordered sequence of items? Go for a list.
- Do you just need to know whether or not you've already *got* a particular value, but without ordering (and you don't need to store duplicates)? Use a set.
- Do you need to associate values with keys, so you can look them up efficiently (by key) later on? Use a dictionary.

answered Aug 15 '10 at 20:24



[Jon Skeet](#)

933k 527 6813
7669

- 14 This should be the "TL;DR" part to the above answer by Alex M. :-) – [Alex Boschmans](#) Feb 23 '15 at 10:42

- 3 I think it is the other way around.. Alex should donate his votes to Jon. This answer covers pretty much everything and it is much more concise and clear. – [mehmet](#) May 2 '16 at 15:59

- Use a dictionary when you have a set of unique keys that map to values.
- Use a list if you have an ordered collection of items.
- Use a set to store an unordered set of items.

answered Aug 15 '10 at 20:25



[SLaks](#)

567k 110 1418
1576

When you want an unordered collection of unique elements, use a `set`. (For example, when you want the set of all the words used in a document).

When you want to collect an immutable ordered list of elements, use a `tuple`. (For example, when you want a (name, phone_number) pair that you wish to use as an element in a set, you would need a tuple rather than a list since sets require elements be immutable).

When you want to collect a mutable ordered list of elements, use a `list`. (For example, when you want to append new phone numbers to a list: [number1, number2, ...]).

When you want a mapping from keys to values, use a `dict`. (For example, when you want a telephone book which maps names to phone numbers: {'John Smith' : '555-1212'}). Note the keys in a dict are unordered. (If you iterate through a dict (telephone book), the keys (names) may show up in any order).

edited Nov 9 '12 at 13:31

answered Aug 15 '10 at 20:25



[unutbu](#)

412k 55 787 900

Uh, he didn't ask about tuples. – [habnabit](#) Aug 15 '10 at 20:29

Although this doesn't cover `set` s, it is a good explanation of `dict` s and `list` s:

Lists are what they seem - a list of values. Each one of them is numbered, starting from zero - the first one is numbered zero, the second 1, the third 2, etc. You can remove values from the list, and add new values to the end. Example: Your many cats' names.

Dictionaries are similar to what their name suggests - a dictionary. In a dictionary, you have an 'index' of words, and for each of them a definition. In python, the word is called a 'key', and the definition a 'value'. The values in a dictionary aren't numbered - they are similar to what their name suggests - a dictionary. In a dictionary, you have an 'index' of words, and for each of them a definition. The values in a dictionary aren't numbered - they aren't in any specific order, either - the key does the same thing. You can add, remove, and modify the values in dictionaries. Example: telephone book.

<http://www.sthurlow.com/python/lesson06/>

edited Aug 19 '15 at 18:43



[Josh Caswell](#)

54k 11 111 159

answered Oct 22 '12 at 0:11



[Goose](#)

356 3 12 27

1 This is an exceptional question, most answerers have ~500k reputation.. – [mehmet](#) May 2 '16 at 16:00
