

Sets

Dictionaries store key-value pairs. You can think of a dictionary as associating a key with a value. The keys, of course, must be unique. There is no order to the dictionary, at least there is no apparent order.

We won't look at the way dictionaries are created (take CSCI 151 for that) but they are structured so that lookup times are very fast.

You can look up a key in a dictionary with n items in time proportional to $\log(n)$; the lookup time for an unordered list of size n is proportional to n .

There is a big difference. $\log_2(1,000,000)$ is about 20. Would you rather do 20 steps or 1,000,000?

Sometimes you want to store data like the keys of a dictionary -- the data values are unique and not ordered; you just want to know if they are there or not and you want to be able to find out quickly.

You could do this with a dictionary -- associate every key with the value None. That seems lame.

Mathematicians call an unordered structure that holds unique objects a *set*. Python has a *set* structure that is implemented in much the same way as a dictionary, only there is no value associated with the keys.

Here are some methods for working with Python sets:

`S = set()` creates a new set

`S = set(L)` creates a set from the elements of list L

`S.add(x)` adds element x to set S

`S.remove(x)` removes element x from S; this throws an error if x is not an element of S

`S <= T` returns True if S is a subset of T; `S >= T` is similar

`x in S` or `x not in S` returns True or False depending on whether set S contains element x

`for x in S` iterates through the elements of S

Here are the Python data structures we have discussed so far:

Lists

Tuples

Dictionaries

Sets

When should you use these?

A) Simplicity:

Tuples are simpler than lists, but you can easily append onto a list and not onto a tuple.

Lists are simpler than dictionaries and sets.

A for-loop can run through any of these structures in the same amount of time

B) Functionality

Lists and tuples are inherently ordered

Sets have faster lookup times than lists or tuples

Dictionaries allow you to associate one data item with another.

Now we have a few clicker questions.

Suppose you want to make a program that will allow me to enter a student's name and have it print out that student's T-number. How would you store the name and T-number data?

- A) As a list
- B) As a pair of lists -- one for the names, one for the T-numbers
- C) As a dictionary
- D) As a set

Suppose you are writing an animation program and you have a large collection of rectangles and circles on the screen. At each step of the animation you need to run through your collection of objects and update the position of each one. How would you store this collection?

- A) As a list
- B) As a dictionary
- C) As a set

Remember our "words25K.txt" file that stores 25,000 words. What kind of structure could you store these words in that would quickly tell you if a given string is one of these words?

- A) A list
- B) A dictionary
- C) A set