Python -sets

Prof. Damitha Karunaratna

University of Colombo school of computing

Types of collections

Python collections can be classified into different groups based on the following properties.

- Sequences
- Mutability
- Iterablility
- Allow replication of elements

How to check whether a collection is iterable? Method 1

- dir()
- Check for the the magic method __iter__

How to check whether a collection is iterable? Method 1

x = {1,2,3}

"__iter__" in dir(x)

"__iter__" in dir(set)

Python hashable objects

An object is hashable if it has a hash value which never changes during its lifetime. All of Python's immutable built-in objects are hashable

By using the function hash(object) you can determine whether an object is hashable or not.

Example :

hash("abcd")

hash([1,2,3])

Properties of python sets

- Mutability
- Iterable
- Replication of elements not allowed

Set construction

Creating a new set

S = {1,2.3,True,"abc"}

Python sets have the following properties:

- They are collections
- elements in a ser are *unordered and hashable.*
- *Existing elements are unchangeable(immutable) but new items can be added and existing items can be removed.*
- *They are unindexed*.
- Sets cannot have duplicate items.
- The items in a set can be of different data types.

Set operations

len(x) – number of item in the set x

Item in x : True only when item is in the set, False otherwise

x.update(tule) – add the elements in the tuple to the set x.

Example:

x.update((5,))

x.update((10,12))

x.remove(item) – removes the element item from the set x if it is in the set, generates an error if item in not in the set x.

x.discard(item) – remove the element item from the set x, but does not generates an error if item in not an element of x

x.clear() – remove all elements from the list x.

Other set operations

first_set.intersection(second_set)

first_set.union(second_set)

first_set.difference(second_set)

a.issubset(b) method or <= operator returns true if the a is a subset of b

a.issuperset(b) method or >= operator returns true if the a is a superset of b

a.isdisjoint(b) method return true if there are no common elements between sets a and b