

Set

Contrasting a set with respect to a list:

List	Ordered	Indexed	Duplicacy allowed	Declared using [and]
Set	Unordered	Unindexed	Duplicacy not allowed	Declared using { and }

A set is a group of items where each item is unique.

It is unordered (means there is no sequence and hence no index).

Note: Ordered means there exists a sequence in the arrangements of grouped items. Unordered means there is no sequence.

Ordered means that the order in which items were placed at the time of creation is maintained by the Python till no modification is done to the data itself. For ex: List and Tuple.

List Methods	Set Methods
append()	add()
extend()	update() [Note: set does not support '+' operator]
remove()	remove(), discard()
pop(): removes the last element by default [Note: also accepts index to pop() an element]	pop(): we cannot say which element will be popped as a set is not ordered.
clear(): empties the list	clear(): empties the set

Problem

What is the output of below code:

```
s1 = {'apple', 'banana', 'carrot'}  
s2 = {'apple', 'pineapple', 'cherry'}
```

```
try:  
    print(s1 + s2)  
except Exception as e:  
    print(e)
```

```
try:  
    print(s1 - s2)  
except Exception as e:  
    print(e)
```

Method	Description
<u>add()</u>	Adds an element to the set
<u>clear()</u>	Removes all the elements from the set
<u>copy()</u>	Returns a copy of the set
<u>difference()</u>	Returns a set containing the difference between two or more sets
<u>difference_update()</u>	Removes the items in this set that are also included in another, specified set
<u>discard()</u>	Remove the specified item
<u>intersection()</u>	Returns a set, that is the intersection of two other sets
<u>intersection_update()</u>	Removes the items in this set that are not present in other, specified set(s)
<u>isdisjoint()</u>	Returns whether two sets have a intersection or not
<u>issubset()</u>	Returns whether another set contains this set or not
<u>issuperset()</u>	Returns whether this set contains another set or not
<u>pop()</u>	Removes an element from the set
<u>remove()</u>	Removes the specified element
<u>symmetric_difference()</u>	Returns a set with the symmetric differences of two sets
<u>symmetric_difference_update()</u>	inserts the symmetric differences from this set and another
<u>union()</u>	Return a set containing the union of sets
<u>update()</u>	Update the set with the union of this set and others

Intersection_update(): keeps the items in this set that are present in the other.