

Q1 What is set in python and how do we enter the set from user at runtime?

A set is a cartesian which is unordered & unindexed. In python sets are written with curly brackets  $\{\}$ .

$S = \{'a', 'b', 'c'\}$

or

$S = \text{set}(['a', 'b', 'c'])$ .

Set is also iterable data type which is mutable in nature and has no duplicate elements.

Q2 How many methods are in set in python. Explain each with examples.

There are total 17 set functions in python.

(i) `add()`: Adds an element to the set.

E.g. `car = {'BMW', 'Rolls-Royce'}`  
`car.add('Lamborghini')`  
`print(car)`

(ii) `clear()`: Removes all elements from the set.

E.g. `car = {'BMW', 'Tesla', 'Mercedes'}`  
`car.clear()`  
`print(car)`

(iii) `copy()`: Returns a copy of the set.

E.g. `l = {'a', 'b', 'c'}`  
`x = l.copy()`  
`print(x)`

- (iv) `difference()`: Returns a set containing the diff between two or more sets  
 E.g.  $x = \{ 'a', 'b', 'c' \}$   
 $y = \{ 'b', 'c', 'd' \}$   
`print(x.difference(y))` →  $\{ 'a' \}$
- (v) `difference_update()`: Removes the items in the set that are also included in another set.  
 E.g.  $x = \{ 'a', 'b', 'c' \}$   
 $y = \{ 'a', 'd', 'b', 'e' \}$   
`print(x.difference_update(y))` →  $\{ 'c', 'd', 'e' \}$
- (vi) `discard()`: Removes the specified items  
 E.g. →  $l = \{ 'a', 'b', 'c', 'd' \}$   
`l.discard('d')`  
`print(l)` →  $\{ 'a', 'b', 'c' \}$
- (vii) `intersection()`: Returns a set that is intersection of two others  
 E.g.  $x = \{ 'Apple', 'Orange', 'Banana' \}$   
 $y = \{ 'Google', 'Microsoft', 'Apple' \}$   
`print(x.intersection(y))` →  $\{ 'Apple' \}$
- (viii) `intersection_update()`: Removes the item in this set that are not present in other specified set/s  
 E.g. →  $S1 = \{ 'a', 'b', 'c', 'd' \}$   
 $S2 = \{ 'b', 'd', 'e', 'f' \}$   
`print(S1.intersection_update(S2))` →  $\{ 'b', 'd' \}$
- (ix) `isdisjoint()`: Returns whether two sets have a intersection or not.  
 E.g. →  $S1 = \{ 'Apple', 'Banana', 'Cherry' \}$   
 $S2 = \{ 'Microsoft', 'Apple', 'Google' \}$   
`print(S1.isdisjoint(S2))` → True



(X) `issubset()`: returns whether another set contains this set or not.

E.g `S1 = {'a', 'b', 'c'}`

`S2 = {'a'}`

`print(S1.issubset(S2))`

(XI) `issuperset()`: Returns whether this set contains another set or not.

E.g. : `S1 = {'a', 'b', 'c'}`

`S2 = {'a'}`

`print(S1.issuperset(S2))`

(XII) `pop()`: removes an element from the set.

E.g: `Cars = {'BMW', 'Mercedes', 'Lamborghini', 'Tesla'}`

`Cars.pop()`

`print(Cars) → {'BMW', 'Mercedes', 'Lamborghini'}`

(XIII) `remove()`: removes the specified element

E.g → `L = {'a', 'b', 'c', 'd'}`

`L.remove('b')`

`print(L) → {'a', 'c', 'd'}`

(XIV) `symmetric_difference()`: returns a set with the symmetric difference of two sets.

E.g: `S1 = {'a', 'b', 'c', 'd'}`

`S2 = {'c', 'd', 'e', 'f'}`

`print(S1.symmetric_difference(S2))`

`→ {'a', 'b', 'e', 'f'}`

(XV) `symmetric_difference_update()`: inserts the symmetric difference from the set & another

Eg: `x = {'A', 'B', 'C'}`

`y = {'a', 'b', 'c'}`

`x.symmetric_difference_update(y)`

`print(x) → {'A', 'B', 'a', 'b'}`

(xvi) `union()`: Returns a set containing the union of sets.

E.g: `x = {'a', 'b', 'c'}`

`y = {'b', 'c', 'd'}`

`print(x.union(y))` → `{'a', 'b', 'c', 'd'}`

(xvii) `update()`: Updates the set with the union of this set & others

E.g: `x = {'Apple', 'Cherry', 'Mango'}`

`y = {'Microsoft', 'Apple', 'Google'}`

`x.update(y)`

`print(x)` → `{'Apple', 'Cherry', 'Mango', 'Microsoft', 'Google'}`

Q3. Is set mutable or immutable? Justify your answer.

Yes, set is mutable since they are unordered, indexing in set have no meaning. We cannot access or change an element of set using indexing or slicing. Slicing does not support indexing.