TUPLES

Tuples are just same as the list, which is used to store multiple value.

Tuples is one of 4 built-in data types in Python used to store collections of data.

It is a collection of data which is ordered, and on unchangeable.

e.g

```
fruits = ("mango", "pineapple")
print(len(fruits))
print(type(fruits))
print(fruits[1])
```

O/P

```
2
<class 'tuple'>
pineapple
```

Once a tuple is created, you cannot change its values. Tuples are unchangeable or Immutable as it also is called.

But there is a workaround. You can convert the tuple into a list, change the list, and convert the list back into a tuple.

```
x = ("apple", "banana", "cherry")
y = list(x)
y[1] = "watermelon"
x = tuple(y)
print(x)
```

fruits = ("apple", "banana", "cherry")

(green, yellow, red) = fruits.

print (green) → apple

print (yellow) → ~~yellow~~ banana

print (red) → ~~re~~ cherry

SETS

These are just like above both, used to store multiple values. Sets are unordered, unindexed, ~~un~~ changeable and it will not allow duplicate.

```
fruits = {"apple", "banana", "cherry"}
  print (fruits)
```

We can use pop(), del(), clear().

Union() function

```
set1 = {"a", "b", "c"}
set2 = [1,2,3]
set3 = set1.union(set2)
print (set3)
```

update()

```
set1 = {"a", "b", "c"}
set2 = {1,2,3}
set1.update(set2)
print(set1)
```

intersection_update()

```
x = {"a", "b", "c"}
y = {"d", "e", ~~}~~"a"}
x.intersection_update(y)
print(x)
```

symmetric_difference_update()

```
set1 = {"a", "b", "c", 1}
set2 = {1, 2, 3}
set1.symmetric_difference_update(set2)
print(set1)
```

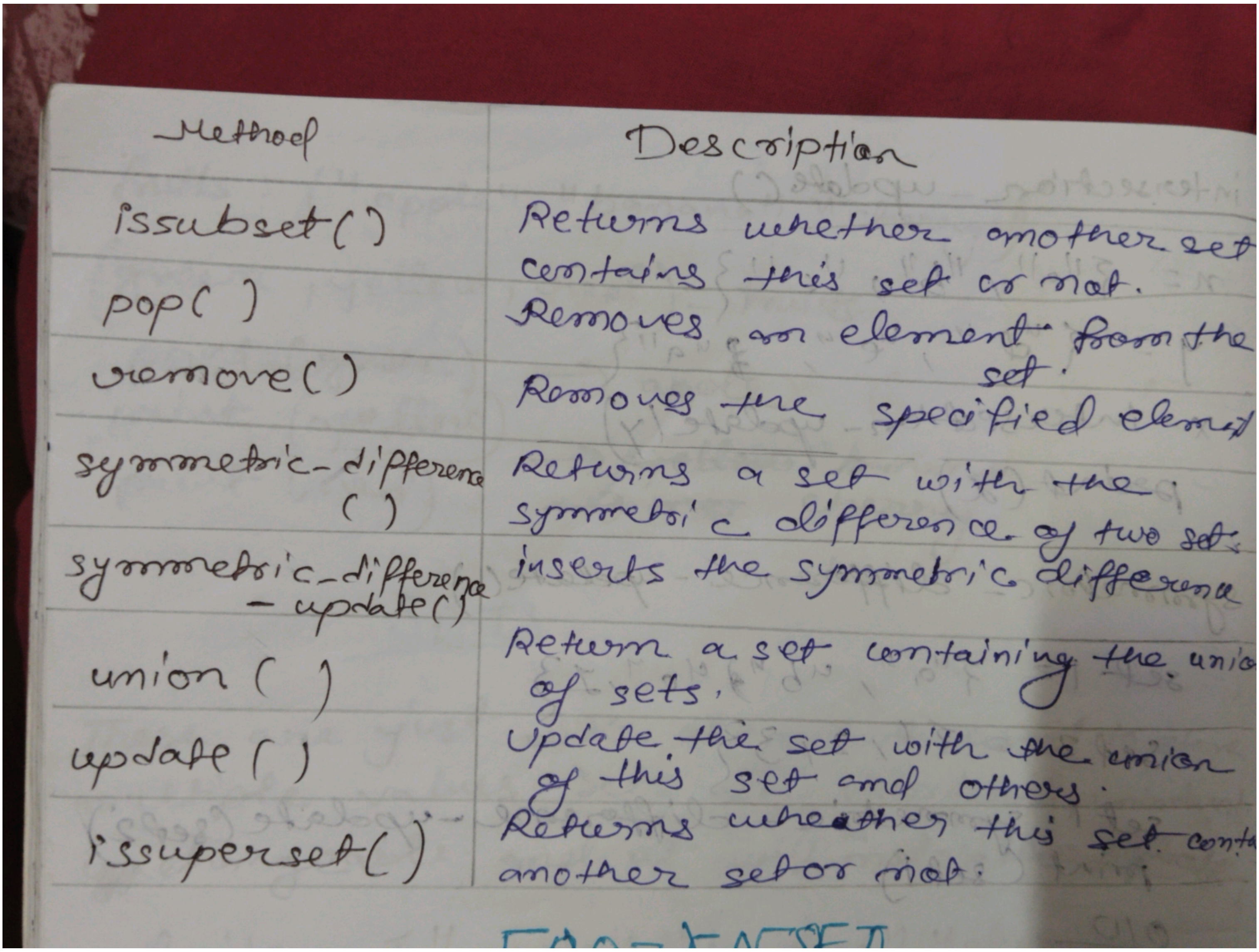
o/p

{'a', 3, 'c', 6, 'b'}

or ~~xoe~~

```
z = x.symmetric_difference(y)
print(z)
```

Method	Description
add()	Adds an element to the set
clear()	Removes all the elements from the set.
copy()	Returns a copy of the set.
difference()	Returns a set containing the difference between two or more sets.
difference_update()	Removes the items in this set that are also included in another, specified set
discard()	Remove the specified item.
intersection	Returns a set, that is the intersection of two other sets
itersection_update	Removes the items in this set that are not present in other, specified sets).
isdisjoint()	Returns whether two sets have a intersection or not.



Method	Description
issubset()	Returns whether another set contains this set or not.
pop( )	Removes an element from the set.
remove()	Removes the specified element
symmetric-difference ( )	Returns a set with the symmetric difference of two sets.
symmetric_difference -update()	inserts the symmetric difference
union ( )	Return a set containing the union of sets.
update ( )	Update the set with the union of this set and others.
issuperset()	Returns wheather this set contains another set or not.

FROZENSET