TUPLE

Ordered, Indexed and Unchangeable. Allows duplicate. Elements can be of any data type.

In [1]:
```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = thistuple.count(5)
print(x)
```

```
2
```

In [2]:
```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
x = thistuple.index(8)  # return first index of occurence. Raises exception if not found
print(x)
```

```
3
```

In [3]:
```
thistuple = ("apple",)
print(type(thistuple))

#NOT a tuple
thistuple = ("apple")
print(type(thistuple))
```

```
<class 'tuple'>
<class 'str'>
```

Updating Tuple

Since they are immutable. Therefore, we first cast them as LIST, then change its contents, and then we'll make them again tuple with changes items.

In [4]:
```
x = ("apple", "banana", "cherry")
y = list(x)
y[1] = "kiwi"
y.remove("apple")
x = tuple(y)
print(x)

del x
try:
    print(x)
except:
    print("Tuple x no longer exists")
```

```
('kiwi', 'cherry')
Tuple x no longer exists
```

HOWEVER

Joining of Tuple can be done easily.

In [5]:
```
tuple1 = ("a", "b", "c")
tuple2 = (1, 2, 3)
tuple3 = tuple1 + tuple2
print(tuple3)

fruits = ("apple", "banana", "cherry")
mytuple = fruits * 2
print(mytuple)
```

```
('a', 'b', 'c', 1, 2, 3)
('apple', 'banana', 'cherry', 'apple', 'banana', 'cherry')
```

In [6]:

```
# n(var on LHS) = n(var on RHS)
# if LHS is tuple, it expects RHS to be tuple.
# if RHS is tuple, LHS may[not] be tuple

fruits = ("apple", "banana", "cherry")
green, yellow, red = fruits
print(green)
print(yellow)
print(red)

print('#'*15)

fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")
green, yellow, *red = fruits
print(green)
print(yellow)
print(red)

print('#'*15)

fruits = ("apple", "mango", "papaya", "pineapple", "cherry")
(green, *tropic, red) = fruits
print(green)
print(tropic)
print(red)
```

```
apple
banana
cherry
###############
apple
banana
['cherry', 'strawberry', 'raspberry']
###############
apple
['mango', 'papaya', 'pineapple']
cherry
```

In [7]:

```
# tuples(or any other sequences) can be compared. Underlying concept is, corresponding elements are compared.

print((0,1,2) < (5,1,2))
print(('jones', 'Sally') < ('Jones', 'Sam'))
print(ord('j'))
print(ord('J'))
```

```
True
False
106
74
```