Its is   an unchangeable.

\#   Allow Duplicate values.

tuple = ('apple', 'banana', 'cherry', banana').

print (tuple)

\# tuple len
     print (len (tuple)).
        ↳ 3.

\# data type.
     print (datatype (tup)).
        ↳ (class tuple).

\# Access tuples a items
tuples = ('apple', 'car', 'bike', 'blue', 89, 54),
print (tuples [2]) → bike
print (tuples [-1]) → 54
print (tuples [2:6]) → bike, blue, 89

Print (tuples [6:-1]) ⟹ (54, 89, 'blue', 'bike', 'car',
                                       'apple')

print (tuples [-4:-1]).
     ↳ ('bike', 'blue', 89]

\# use if modal.
tuples = ('apple', 'car', 'bike', 'blue', 89, 54).

     If 'apple' in tuples :
        print (Yes, apple is there in.
                tuples).

    else :
      print(No)

data type.

```
my tuple = ('apple', 'blue').
    print (type (my tuple))
                ↳ ('clam tuple).


    tuple = ('a')
        Print (type (tuple)) → ('clam string).
```

※ we get the index value mean will use
[] (square bracket).

※ we can't add any value in tuple and
we can't delete the value but we can
delete hole tuple.
    E.X

Print (del my tuple ()).
    ↳ It will be delete.

modification.
    we can't change only tuple value not
a list. here the list List inside the tuple.

E.X   my tup = (20, 25, 30 ['apple', 'orange'], 57)

        my tup [3] [0] = "kiwi"

        print (my tup)
            ↳ (20,25,30 ['kiwi', 'orange'],57].
                            ↓
                    now we get the change
                    in list value.

Concatenation

we can add two tuple but we can't add and modify inside the tuple.

```
tuple1 = (1, 3, 5, 7, 9).
tuple2 = (7, 8, 29)

tuple = tuple1 + tuple2
    └ print (tuple).
            └ (1, 3, 5, 7, 9, 7, 8, 29).
```

replication.

get the two value.

```
tuple = (1, 3, 5)
print (tuple*2)
    └ (1, 3, 5, 1, 3, 5).
```

method of in tuples
1. Count (element).
2. index (element).

count.

```
tuple = (1, 2, 3, 2, 5, 2)
print = (tuple. count (2))
        ⇒ 3.
```

index value.

```
print
print (tuple.index (1))
    └ 0.
        └ get the index position of the value.
```

```
# Loop Through a tuple  (for).
    this tuple = ('apple', 'banana', 'bike')
    for x in thistuple:
        print (x)
            ⌐> apple
            banana
            bike
```

```
Loop through   a tuple (while).
    tuple = ('car', 'bike', 'vichele')
    i = 0
    while i < len (tuple):
        print (tuple [i])
        i = i + 1
            ⌐> print  car
                bike
                vichele.
```