Tuple is one of the built-in data type and return in () ,used to store multiple items in single variable.

Tuple is collection which is ordered and unchangeable

Tuple allowed duplicate value with any data type (int,string,boolean)

To create tuple with one items:-madd a comma after the item otherwise it will show error

**Creating tuple**

With the help of constructor:-

Tuple1=tuple((1,2,3,4))

**Access tuple item**

Access tuple items by referring to the index number,inside [] brackets

Tuple1=(1,2,3,4,5,6)

Print(Tuple1[3])>>>4

Print(Tuple1[-1]) >>>6

**Checking exists**

Tuple1=(1,2,3,4,5,6)

If 4 in Tuple:

Print (“yes”)

yes

Methods

1.count(value): Returns the number of occurrences of a specified value in the tuple.

my\_tuple = (1, 2, 3, 4, 1, 1)

print(my\_tuple.count(1)) # Output: 3

2.index(value, start=0, stop=len(tuple)): Returns the index of the first occurrence of a specified value in the tuple. Optionally, you can specify the start and stop indices for the search.

my\_tuple = ('a', 'b', 'c', 'd', 'e')

print(my\_tuple.index('c')) # Output: 2

Operations:

1.Accessing Elements: Elements of a tuple can be accessed using indexing or slicing.

my\_tuple = (10, 20, 30, 40)

print(my\_tuple[0]) # Output: 10

print(my\_tuple[1:3]) # Output: (20, 30)

2.Concatenation: Tuples can be concatenated using the + operator.

tuple1 = (1, 2, 3)

tuple2 = (4, 5, 6)

concatenated\_tuple = tuple1 + tuple2

print(concatenated\_tuple) # Output: (1, 2, 3, 4, 5, 6)

3.Repetition: Tuples can be repeated using the \* operator.

my\_tuple = (1, 2)

repeated\_tuple = my\_tuple \* 3

print(repeated\_tuple) # Output: (1, 2, 1, 2, 1, 2)

function

In Python, tuples are immutable sequences, and they come with a variety of built-in functions and operations to manipulate and work with them. While tuples themselves don't have as many functions as lists, there are several functions that are commonly used with tuples. Here are some functions that are frequently used with tuples:

1.len(tuple): Returns the number of elements in the tuple.

my\_tuple = (1, 2, 3, 4, 5)

print(len(my\_tuple)) # Output: 5

2.max(tuple): Returns the largest element in the tuple.

my\_tuple = (10, 30, 20, 40)

print(max(my\_tuple)) # Output: 40

3.min(tuple): Returns the smallest element in the tuple.

my\_tuple = (10, 30, 20, 40)

print(min(my\_tuple)) # Output: 10

4.sum(tuple): Returns the sum of all elements in the tuple (only applicable if elements are numeric)

my\_tuple = (1, 2, 3, 4, 5)

print(sum(my\_tuple)) # Output: 15

5.sorted(tuple): Returns a new sorted list from the elements of the tuple.

my\_tuple = (10, 5, 20, 15)

sorted\_tuple = sorted(my\_tuple)

print(sorted\_tuple) # Output: [5, 10, 15, 20]

tuple are unchangeable ,meaning that you cannot change add or remove items once the tuple is created but there are some workaround

**1.change tuple values**

Onces a tuple is created you cannot change its value but you can convert the tuple into list make operation and convert back list into tuple

e.g. x=(“a”,”b”,”c”)

y=list(x)

y[1]=”A”

x=tuple(y)

**2.unpacking a tuple**

Assigning value is called packing and like this extract the value back into variable is called unpacking

my\_tuple = ('a', 'b', 'c', 'd', 'e')

(A,B,C,D,E)=my\_tuple

Print(A) >>a

The no of variable must match the number of value in the tuple if not thrn you must use an asterisk to collect remaning value as a list . When no of variable is less then you can add\*to the no

my\_tuple = ('a', 'b', 'c', 'd', 'e')

(A,B,\*E)=my\_tuple

Print(E) >>(c,d,e)

**3.Loops in tuple**

1. For loop

my\_tuple = ('a', 'b', 'c', 'd', 'e')

for x in my\_tuple:

print(x)

for I in range(len(my\_tuple)):

print(my\_tuple(i))

1. while loop

my\_tuple = ('a', 'b', 'c', 'd', 'e')

i=0

while i<len(my\_tuple):

print(my\_tuple)

i=i+1