Tuple Concepts

1.How to read tuple

- Parenthesis () symbol.
- Predefind tuple(p) function.
- A tuple can store group of objects or elements.
- A tuple can store same (Homogeneous) type of elements.
- A tuple can store different (Heterogeneous) type of elements.
- In tuple insertion order is preserved of fixed.

In [1]:
```
tuple=('1,2,3')
print(tuple)
```

```
1,2,3
```

- Duplicate elements are allowed.
- Tuple having immutable nature.
- Store elements by using index.
- A tuple data structure supports both posetive and negative indexes.
- Posetive index means from left to right.
- Negative index means from right to left.
- Tuple is predefind class in python.
- Inside tuple ever object can be seperated by comma separator.

In [3]:
```
# tuple having same type of objects

numbers=(100,200,300,400,500)
print(numbers)

print(type(numbers))
```

```
(100, 200, 300, 400, 500)
<class 'tuple'>
```

In [4]:
```
# A single value with tuple syntax, but its not tuple

number=(10)
print(number)
print(type(number))
```

```
10
<class 'int'>
```

In [5]:
```
name=("Aviansh")
print(name)
print(type(name))
```

```
Aviansh
<class 'str'>
```

In [6]:
```
name=("Avinash",)
print(name)
print(type(name))
```

```
('Avinash',)
<class 'tuple'>
```

- Parenthesis is optional for tuple
- While creating a tuple paraenthesis is optional

In [7]:
```
number=100,200,300,400,500
print(number)
```

```
(100, 200, 300, 400, 500)
```

2.Diffrent way to create a tuple

a. empty tuple

In [9]:
```
number=()
print(number)
print(type(number))
```

```
()
<class 'tuple'>
```

b. tuple with group of values

In [10]:
```
number=(10,20,30)
number1=100,200,300
print(number)
print(number1)
```

```
(10, 20, 30)
(100, 200, 300)
```

In [11]:
```
a=(10,20,30,'avinash')
print(a)
```

```
(10, 20, 30, 'avinash')
```

we can access tuple elements by using

- index
- slice

index

- index means position where element store

In [22]:
```
a=(10,20,30,40,50,60,70,80,90,100)
print(a[0])
print(a[-1])
```

```
10
100
```

slice operator

- A group of objects from starting point to ending point.

In [23]:
```
a=(10,20,30,40,50,60,70,80,90,100)
print(a[2:6])
print(a[2:100])
print(a[::3])
```

```
(30, 40, 50, 60)
(30, 40, 50, 60, 70, 80, 90, 100)
(10, 40, 70, 100)
```

3. tuple vs immutability

- tuple having immutable nature.
- if we create a tuple than we cannot modify the elements of existing tuple.

In [24]:
```
a=(10,20,30,40)
print(a[1])
a[1]=50
```

```
20

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call las
t)
Cell In[24], line 3
      1 a=(10,20,30,40)
      2 print(a[1])
----> 3 a[1]=50

TypeError: 'tuple' object does not support item assignment
```

4. Mathematical operators on tuple:

- we can apply plus(+) and multiplication (*) operators on tuple.
- (+) operator works a sconcatination.
- (-) operator works a multiplication.

a. concatination operator (+):

- (+) operator concatination two tuples and returns single tuple

In [25]:
```
a1=(10,20,30)
a2=(40,50,60)
a3=a1+a2
print(a3)
```

```
(10, 20, 30, 40, 50, 60)
```

- multiplication operator works as repetition operator

In [27]:
```
a1=(10,20,30)
a3=a1*4
print(a3)
```

```
(10, 20, 30, 10, 20, 30, 10, 20, 30, 10, 20, 30)
```

- len(p) function

In [28]:
```
a=(10,20,30,40)
print(len(a))
```

```
4
```

5. method in tuple data structure

- as discussed , tuple is a predefiend class.
- so,tuple class can contain method because method can be created inside of class only.
- we can check these methods by using dir(p) predefinend function.
- so, internally tuple class contains two types of method:
  - with underscore symbol method:
    - we no need to focus.
  - without underscore symbol method:
    - we need to focus much on these.

In [29]:
```
print(dir(tuple))
```

```
['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc_
_', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__',
'__getnewargs__', '__getstate__', '__gt__', '__hash__', '__init__', '__ini
t_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__mod__', '__mu
l__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__r
mod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshoo
k__', 'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith',
'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalph
a', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumer
ic', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'low
er', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix',
'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip',
'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'transl
ate', 'upper', 'zfill']
```

Importent point

- As per object-oriented principle:
  - if we want to acces instance method than we should access by using object name.
  - so, all tuple method we can access by using tuple object.

Methods in tuple

$1.count(parameter1)method$

$2.index(parameter1)method$

count(p) method

- count(p) is a method , we should acess this method by using tuple object.
- this method return the number of occurrences of specified item in the tuple.

In [31]:
```
a=(10,20,10,10,20)
print(a.count(10))
```

```
3
```

$index(p)method$

- return index of first of the given element.
- if the specified element is not available,then we will get valueError.

In [32]:
```
a=(10,20,30)
print(a.index(30))
```

```
2
```

In [33]:
```
a=(10,20,30)
print(a.index(80))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call las
t)
Cell In[33], line 2
      1 a=(10,20,30)
----> 2 print(a.index(80))

ValueError: tuple.index(x): x not in tuple
```

Can i add elements to this tuple t=(111,222,[333,444],555,666)

- yes we can add elements to list in tuple

In [34]:
```
b=(111,222,[333,444],555,666)
b[2].append(777)
print(b)
```

```
(111, 222, [333, 444, 777], 555, 666)
```

Thankyou !