In [1]:
```
# define tuple
tup=()
type(tup)
```

Out[1]:
```
tuple
```

In [2]:
```
# tuple with single element
tup=tuple('12',)
type(tup)
```

Out[2]:
```
tuple
```

In [3]:
```
t=tuple()
type(t)
```

Out[3]:
```
tuple
```

In [ ]:
```

```

In [4]:
```
t=(23)
print(t)
type(t)
```

```
23
```

Out[4]:
```
int
```

In [5]:
```
tup = ("apple")
print(type(tup))
```

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

In [6]:
```
t=(23,)
print(t)
type(t)
```

```
(23,)
```

Out[6]:
```
tuple
```

In [7]:
```
t=()
print(type(t))
```

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

In [8]:
```
color=("red","blue","yellow","pink")
print(color)
```

```
('red', 'blue', 'yellow', 'pink')
```

In [9]:
```
# length of tuple
len(color)
```

Out[9]:
```
4
```

In [11]:
```
# acces elemet from tuple
print(color[0])
```

```
red
```

In [12]:
```
print(color[-2:-1])
```

```
('yellow',)
```

In [15]:
```
print(color[-3])
```

```
blue
```

In [18]:
```
print(color[-1])
```

```
pink
```

In [19]:
```
tup=("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
# return the third, fourth, and fifth item
print(tup[2:5])
```

```
('cherry', 'orange', 'kiwi')
```

In [21]:
```
print(tup[-5:-2])
```

```
('cherry', 'orange', 'kiwi')
```

In [22]:
```
for x in tup:
    print(x,end=" ")
```

```
apple banana cherry orange kiwi melon mango
```

In [23]:
```
# Once a tuple is created, you cannot change its values. Tuples are unchangeable, or immutable as it also is called
```

In [24]:
```
tup.update(23)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[24], line 1
----> 1 tup.update(23)

AttributeError: 'tuple' object has no attribute 'update'
```

In [25]:
```
tup.append("pink")
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[25], line 1
----> 1 tup.append("pink")

AttributeError: 'tuple' object has no attribute 'append'
```

In [26]:
```
tup.remove("apple")
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[26], line 1
----> 1 tup.remove("apple")

AttributeError: 'tuple' object has no attribute 'remove'
```

In [27]:
```
# convert tuple in list
l=list(tup)
print(l)
```

```
['apple', 'banana', 'cherry', 'orange', 'kiwi', 'melon', 'mango']
```

In [28]:
```
l.append("pink")
```

In [29]:
```
tup=tuple(l)
tup
```

Out[29]:
```
('apple', 'banana', 'cherry', 'orange', 'kiwi', 'melon', 'mango', 'pink')
```

In [30]:
```
# count elements in tuple
tup=(12,23,23,43,32,12,12)
tup.count(12)
```

Out[30]:
```
3
```

In [44]:
```
tup=(12,23,23,43,32,12,12)
# for i in range(len(tup)):
for x in tup:
    c=tup.count(x)
    print(f"count of {x} is : {c}")
```

```
count of 12 is : 3
count of 23 is : 2
count of 23 is : 2
count of 43 is : 1
count of 32 is : 1
count of 12 is : 3
count of 12 is : 3
```

In [45]:
```
# Add tuple to a tuple
seas=("rainy", "winter")
sea=("summer",)
seasons=seas+sea
print(seasons)
```

```
('rainy', 'winter', 'summer')
```

In [46]:
```
# When we create a tuple, we normally assign values to it. This is called "packing" a tuple
print(seasons)
```

```
('rainy', 'winter', 'summer')
```

In [48]:
```
# we are also allowed to extract the values back into variables. This is called "unpacking
```

In [49]:
```
tup=("English","red","Lotus")
(language,color, flower)=tup
print(language)
print(color)
print(flower)
```

```
English
red
Lotus
```

In [50]:
```
# If the number of variables is less than the number of values, we
# can add an * to the variable name and the values will be assigned to the variable as a list
tup=("English","Hindi","Marathi","red","Lotus")
(*language,color, flower)=tup
print(language)
print(color)
print(flower)
```

```
['English', 'Hindi', 'Marathi']
red
Lotus
```

In [51]:
```
# join tuples
t1=(1,2,3,4)
t2=(2,3,4,5)
t=t1+t2
print(t)
```

```
(1, 2, 3, 4, 2, 3, 4, 5)
```

In [52]:
```
t.count(2)
```

Out[52]:
```
2
```

In [54]:
```
t.index(2,2)
```

Out[54]:
```
4
```

In [57]:
```
t1=(1,2,3,4)
t2=(2,3,4,5,2,4,6)
t=t1+t2
print(t)
```

```
(1, 2, 3, 4, 2, 3, 4, 5, 2, 4, 6)
```

In [62]:
```
t.index(2,5)
```

Out[62]:
```
8
```

In [66]:
```
# index() With Start and End Parameters
t.index(2,3,9)
```

Out[66]:
```
4
```

In [67]:
```
# sum of elements of tuple
sum=0
for x in t:
    sum+=x
print(sum)
```

```
36
```

In [68]:
```
tup=([1,2,3],[4,5,6],[6,7,8],[1,2,3])
print(tup)
```

```
([1, 2, 3], [4, 5, 6], [6, 7, 8], [1, 2, 3])
```

In [71]:
```
# remove duplicate list frtom tuple
s=[]
for x in tup:
    if x not in s:
        s.append(x)
print(s)
```

```
[[1, 2, 3], [4, 5, 6], [6, 7, 8]]
```

In [78]:
```
tup=([1,"riya",23],[2,"seeta",26],[3,"Komal",18],[4,"Annu",23])
d=[]
for x in tup:
    d.append({'key':x[0],"name":x[1],"age":x[2]})
print(d)
```

```
[{'key': 1, 'name': 'riya', 'age': 23}, {'key': 2, 'name': 'seeta', 'age': 26}, {'key': 3, 'name': 'Komal', 'age': 18}, {'key': 4, 'name': 'Annu', 'age': 23}]
```

In [ ]:
```
```