# Tupple and Its Functions

In [8]:
```
t1=()  # empty tupple
print(t1)
t2=(1)
print(t2)
t=(1,2,5,8,4,6,"nikhil","vivek")
print(t)
print(id(t))
print(type(t))
```

```
()
1
(1, 2, 5, 8, 4, 6, 'nikhil', 'vivek')
2783256325792
<class 'tuple'>
```

In [3]:
```
t[6][1]
```

Out[3]:
```
'i'
```

In [4]:
```
len(t)
```

Out[4]:
```
8
```

In [11]:
```
tup=(1,7,3,1,2,2,2,2,2,6,"nikhil")
tup.count(2)
```

Out[11]:
```
5
```

In [12]:
```
tup.index(2)  # it returns from first index where the element was found
```

Out[12]:
```
4
```

In [14]:
```
t1=(2,4,6)
t2=(5,66)
print(t1+t2)
print(t2+t1)
```

```
(2, 4, 6, 5, 66)
(5, 66, 2, 4, 6)
```

In [16]:
```
t1=(2,4,6)
t2=(5,66)
print(t2*2)
print(t1*t2)   # throws an error
```

```
(5, 66, 5, 66)

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[16], line 4
      2 t2=(5,66)
      3 print(t2*2)
----> 4 print(t1*t2)

TypeError: can't multiply sequence by non-int of type 'tuple'
```

# Tupple Packing and Unpacking

In [17]:
```
t=1,2,3,4,5,"vivek",75.6
print(t)
print(type(t))
```

```
(1, 2, 3, 4, 5, 'vivek', 75.6)
<class 'tuple'>
```

In [20]:
```
t=1,2,3,4,5,"n"  # tupple packing
a,b,c,d,e,f=t           # tupple unpacking
print(t)
print(type(t))
print(a,b,c,d,e,f)
print(type(a))
print(type(f))
```

```
(1, 2, 3, 4, 5, 'n')
<class 'tuple'>
1 2 3 4 5 n
<class 'int'>
<class 'str'>
```

In [22]:
```
# tupple comprehension is not possible in tupple
x=(i for i in range(1,10)) # do not generate tupple
print(x)
print(type(x))
```

```
<generator object <genexpr> at 0x00000288057F98A0>
<class 'generator'>
```

# Set and its Functions

In [23]:
```
# Empty set is not allowed in python
s={}
print(s)
print(type(s))
```

```
{}
<class 'dict'>
```

In [26]:
```
s={1,23,54,6,15, "nikhil",56,1,0,25}
print(s)
print(type(s))
```

```
{0, 1, 6, 15, 54, 23, 'nikhil', 25, 56}
<class 'set'>
```

In [27]:
```
p={1,2,3,4,6}
q={4,9,1,0,5}
print(p.intersection(q))  # common variables
print(p.union(q))
print(p.difference(q))
```

```
{1, 4}
{0, 1, 2, 3, 4, 5, 6, 9}
{2, 3, 6}
```

In [28]:
```
p.add("sistec")
print(p)
```

```
{1, 2, 3, 4, 6, 'sistec'}
```

In [29]:
```
x={i for i in range(1,11) if i%2==0}
print(x)
```

```
{2, 4, 6, 8, 10}
```

# Dictionary and its Functions

In [31]:
```
d={"key":"value"}
print(d)
print(id(d))
print(type(d))
```

```
{'key': 'value'}
2783270690496
<class 'dict'>
```

In [45]:
```
d={1:"nikhil",2:"sandeep",3:"ankit",4:"Rahul"}
print(d)
print(id(d))
print(type(d))
```

```
{1: 'nikhil', 2: 'sandeep', 3: 'ankit', 4: 'Rahul'}
2783270691968
<class 'dict'>
```

In [46]:
```
print(d[0])
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[46], line 1
----> 1 print(d[0])

KeyError: 0
```

In [47]:
```
print(d[1])  # key is the index
```

```
nikhil
```

In [48]:
```
d.keys()
print(d)
print(d.keys())
print(type(d.keys()))
```

```
{1: 'nikhil', 2: 'sandeep', 3: 'ankit', 4: 'Rahul'}
dict_keys([1, 2, 3, 4])
<class 'dict_keys'>
```

In [49]:
```
d.values()
```

Out[49]:
```
dict_values(['nikhil', 'sandeep', 'ankit', 'Rahul'])
```

In [50]:
```
d.get(2)
```

Out[50]:
```
'sandeep'
```

In [51]:
```
print(d.items())
```

```
dict_items([(1, 'nikhil'), (2, 'sandeep'), (3, 'ankit'), (4, 'Rahul')])
```

In [52]:
```
d.pop(2)  # delete the the value of that key
```

Out[52]:
```
'sandeep'
```

In [53]:
```
d.pop("a")
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[53], line 1
----> 1 d.pop("a")

KeyError: 'a'
```

In [54]:
```
d.clear()  # clear all the elements of dictionary and return empty dictionary
```

In [55]:
```
print(d)
print(type(d))
```

```
{}
<class 'dict'>
```

In [57]:
```
l=["nikhil","monu","sonu"]
d=dict(l)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[57], line 2
      1 l=["nikhil","monu","sonu"]
----> 2 d=dict(l)

ValueError: dictionary update sequence element #0 has length 6; 2 is required
```

In [58]:
```
d1={1:{"n":"nikhil"},2:{"b":"bittu"}}  # nesting of dictionary
print(d1)
```

```
{1: {'n': 'nikhil'}, 2: {'b': 'bittu'}}
```

In [ ]: