In [1]:
```
print("TUPLE ")
tuple_1 = (12, 87, 65, 655, 90, 83)
tuple_2 = (12.09, 87.65, 65.07, 655.12, 90.24, 83.76)
tuple_3 = ("A", "B", "C", "D", "E", "F")
tuple_4 = ("Sharmin", "Anaya", "Yumna", "Ridha")
tuple_5 = (True, False, True, False)
tuple_6 = ("Sharmin", 40, True, 1.09)
tuple_7 = ("Sharmin",)


print("TUPLE OF INTEGER\t\t=", tuple_1)
print("TUPLE OF FLOAT\t\t\t=", tuple_2)
print("TUPLE OF CHARACTER\t\t=", tuple_3)
print("TUPLE OF STRING\t\t\t=", tuple_4)
print("TUPLE OF BOOLEAN\t\t=", tuple_5)
print("TUPLE OF DIFFIERENT DATATYPE\t=", tuple_6)
print("TUPLE OF ONE ELEMENT\t\t=", tuple_7)
```

```
TUPLE
TUPLE OF INTEGER                = (12, 87, 65, 655, 90, 83)
TUPLE OF FLOAT                  = (12.09, 87.65, 65.07, 655.12, 90.24, 83.76)
TUPLE OF CHARACTER              = ('A', 'B', 'C', 'D', 'E', 'F')
TUPLE OF STRING                 = ('Sharmin', 'Anaya', 'Yumna', 'Ridha')
TUPLE OF BOOLEAN                = (True, False, True, False)
TUPLE OF DIFFIERENT DATATYPE    = ('Sharmin', 40, True, 1.09)
TUPLE OF ONE ELEMENT            = ('Sharmin',)
```

In [2]:
```
print("TUPLE - USING CONSTRUCTOR ")
tuple_1 = tuple((12, 87, 65, 655, 90, 83))
tuple_2 = tuple((12.09, 87.65, 65.07, 655.12, 90.24, 83.76))
tuple_3 = tuple(("A", "B", "C", "D", "E", "F"))
tuple_4 = tuple(("Sharmin", "Anaya", "Yumna", "Ridha"))
tuple_5 = tuple((True, False, True, False))
tuple_6 = tuple(("Sharmin", 40, True, 1.09))
tuple_7 = tuple("sharmin")



print("TUPLE OR INTEGER\t\t=", tuple_1)
print("TUPLE OR FLOAT\t\t\t=", tuple_2)
print("TUPLE OR CHARACTER\t\t=", tuple_3)
print("TUPLE OR STRING\t\t\t=", tuple_4)
print("TUPLE OR BOOLEAN\t\t=", tuple_5)
print("TUPLE OR DIFFIERENT DATATYPE\t=", tuple_6)
print("TUPLE OF ONE ELEMENT\t\t=", tuple_7)
```

```
TUPLE - USING CONSTRUCTOR
TUPLE OR INTEGER                = (12, 87, 65, 655, 90, 83)
TUPLE OR FLOAT                  = (12.09, 87.65, 65.07, 655.12, 90.24, 83.76)
TUPLE OR CHARACTER              = ('A', 'B', 'C', 'D', 'E', 'F')
TUPLE OR STRING                 = ('Sharmin', 'Anaya', 'Yumna', 'Ridha')
TUPLE OR BOOLEAN                = (True, False, True, False)
TUPLE OR DIFFIERENT DATATYPE    = ('Sharmin', 40, True, 1.09)
TUPLE OF ONE ELEMENT            = ('s', 'h', 'a', 'r', 'm', 'i', 'n')
```

In [3]:
```
print("TUPLE - LENGTH & TYPE ")

tuple_1 = ("Sharmin", "Anaya", "Yumna", "Ridha")
tuple_2 = ("Sharmin", 40, True, 1.09)
tuple_3 = ("Sharmin",)



print("\nTUPLE OF STRING\t\t\t=", tuple_1)
print("LENGTH OF STRING'S TUPLE\t=", len(tuple_1))
print("TYPE\t\t\t\t=", type(tuple_1))
print("\n")

print("TUPLE OF DIFFIERENT DATATYPE\t=", tuple_2)
print("LENGTH OF THE TUPLE\t\t=", len(tuple_2))
print("TYPE\t\t\t\t=", type(tuple_2))
print("\n")

print("TUPLE OF ONE ELEMENT\t\t=", tuple_3)
print("LENGTH OF THE TUPLE\t\t=", len(tuple_3))
print("TYPE\t\t\t\t=", type(tuple_3))
print("\n")
```

```
TUPLE - LENGTH & TYPE

TUPLE OF STRING                 = ('Sharmin', 'Anaya', 'Yumna', 'Ridha')
LENGTH OF STRING'S TUPLE        = 4
TYPE                            = <class 'tuple'>


TUPLE OF DIFFIERENT DATATYPE    = ('Sharmin', 40, True, 1.09)
LENGTH OF THE TUPLE             = 4
TYPE                            = <class 'tuple'>


TUPLE OF ONE ELEMENT            = ('Sharmin',)
LENGTH OF THE TUPLE             = 1
TYPE                            = <class 'tuple'>
```

In [4]:
```
print("TUPLE - ACCESSING VALUE USING INDEX")

tuple_1 = ("Mango", "Cherry", "Berry", "Apple")

print("\nTUPLE1\t\t\t=", tuple_1)

print("FIRST ELEMENT OF TUPLE\t=",tuple_1[0])
print("SECOND ELEMENT OF TUPLE\t=",tuple_1[1])
print("THIRD ELEMENT OF TUPLE\t=",tuple_1[2])
print("FOURTH ELEMENT OF TUPLE\t=",tuple_1[3])
```

```
TUPLE - ACCESSING VALUE USING INDEX

TUPLE1                  = ('Mango', 'Cherry', 'Berry', 'Apple')
FIRST ELEMENT OF TUPLE  = Mango
SECOND ELEMENT OF TUPLE = Cherry
THIRD ELEMENT OF TUPLE  = Berry
FOURTH ELEMENT OF TUPLE = Apple
```

In [5]:
```
print("TUPLE - ACCESSING VALUE USING INDEX (NEGATIVE)")

tuple_1 = ("Mango", "Cherry", "Berry", "Apple")

print("\nTUPLE1\t\t\t=", tuple_1)

print("FIRST ELEMENT OF TUPLE\t=",tuple_1[-1])
print("SECOND ELEMENT OF TUPLE\t=",tuple_1[-2])
print("THIRD ELEMENT OF TUPLE\t=",tuple_1[-3])
print("FOURTH ELEMENT OF TUPLE\t=",tuple_1[-4])
```

```
TUPLE - ACCESSING VALUE USING INDEX (NEGATIVE)

TUPLE1                  = ('Mango', 'Cherry', 'Berry', 'Apple')
FIRST ELEMENT OF TUPLE  = Apple
SECOND ELEMENT OF TUPLE = Berry
THIRD ELEMENT OF TUPLE  = Cherry
FOURTH ELEMENT OF TUPLE = Mango
```

In [6]:
```
print("TUPLE - ACCESSING VALUE USING INDEX (RANGE)")

tuple_1 = ("Mango", "Cherry", "Berry", "Apple", "Watermelon", "Strawberry", "P

print("\nTUPLE 1\t\t\t\t=", tuple_1)

print("tuple[3:-2] ELEMENT OF TUPLE\t=",tuple_1[3:-2])
print("tuple[4:1] ELEMENT OF TUPLE\t=",tuple_1[4:1])
print("tuple[2:5] ELEMENT OF TUPLE\t=",tuple_1[2:5])
print("tuple[:5] ELEMENT OF TUPLE\t=",tuple_1[:5])
print("tuple[2:] ELEMENT OF TUPLE\t=",tuple_1[2:])
print("\n")


print("tuple[-3:4] ELEMENT OF TUPLE\t=",tuple_1[-3:4])
print("tuple[-3:14] ELEMENT OF TUPLE\t=",tuple_1[-3:14])
print("tuple[-6:5] ELEMENT OF TUPLE\t=",tuple_1[-6:5])
print("tuple[-5:-2] ELEMENT OF TUPLE\t=",tuple_1[-5:-2])
print("tuple[-1:-3] ELEMENT OF TUPLE\t=",tuple_1[-1:-3])
print("tuple[:-2] ELEMENT OF TUPLE\t=",tuple_1[:-2])
print("tuple[-5:] ELEMENT OF TUPLE\t=",tuple_1[-5:])
print("tuple[:] ELEMENT OF TUPLE\t=",tuple_1[:])
```

```
TUPLE - ACCESSING VALUE USING INDEX (RANGE)

TUPLE 1                         = ('Mango', 'Cherry', 'Berry', 'Apple', 'Wate
rmelon', 'Strawberry', 'Pomegranate')
tuple[3:-2] ELEMENT OF TUPLE    = ('Apple', 'Watermelon')
tuple[4:1] ELEMENT OF TUPLE     = ()
tuple[2:5] ELEMENT OF TUPLE     = ('Berry', 'Apple', 'Watermelon')
tuple[:5] ELEMENT OF TUPLE      = ('Mango', 'Cherry', 'Berry', 'Apple', 'Wate
rmelon')
tuple[2:] ELEMENT OF TUPLE      = ('Berry', 'Apple', 'Watermelon', 'Strawberr
y', 'Pomegranate')


tuple[-3:4] ELEMENT OF TUPLE    = ()
tuple[-3:14] ELEMENT OF TUPLE   = ('Watermelon', 'Strawberry', 'Pomegranate')
tuple[-6:5] ELEMENT OF TUPLE    = ('Cherry', 'Berry', 'Apple', 'Watermelon')
tuple[-5:-2] ELEMENT OF TUPLE   = ('Berry', 'Apple', 'Watermelon')
tuple[-1:-3] ELEMENT OF TUPLE   = ()
tuple[:-2] ELEMENT OF TUPLE     = ('Mango', 'Cherry', 'Berry', 'Apple', 'Wate
rmelon')
tuple[-5:] ELEMENT OF TUPLE     = ('Berry', 'Apple', 'Watermelon', 'Strawberr
y', 'Pomegranate')
tuple[:] ELEMENT OF TUPLE       = ('Mango', 'Cherry', 'Berry', 'Apple', 'Wate
rmelon', 'Strawberry', 'Pomegranate')
```

In [7]:
```
print("TUPLE - CHECKING ELEMENT'S EXISTENCE")

tuple_1 = ("Mango", "Cherry", "Berry", "Apple", "Watermelon", "Strawberry", "P
print("\nTUPLE 1\t=", tuple_1)

if "Mango" in tuple_1:
    print("\n\"Mango\" is exist in list.")
else:
    print("\n\"Mango\" is not exist in list.")


if "Banana" in tuple_1:
    print("\n\"Banana\" is exist in list.")
else:
    print("\n\"Banana\" is not exist in list.")


if "Litchi" not in tuple_1:
    print("\n\"Litchi\" is not exist in list.")
else:
    print("\n\"Litchi\" is  exist in list.")


if "Cherry" not in tuple_1:
    print("\n\"Cherry\" is not exist in list.")
else:
    print("\n\"Cherry\" is  exist in list.")
```

```
TUPLE - CHECKING ELEMENT'S EXISTENCE

TUPLE 1 = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawberry',
'Pomegranate')

"Mango" is exist in list.

"Banana" is not exist in list.

"Litchi" is not exist in list.

"Cherry" is  exist in list.
```

In [9]:
```
print("TUPLE - CHANGING, ADDING, DELETING ELEMENT USING LIST")

tuple_1 = ("Mango", "Cherry", "Berry", "Apple", "Watermelon", "Strawberry", "P
print("\nTUPLE\t=", tuple_1)

#CONVERTING TUPLE INTO LIST
list_1 = list(tuple_1)
print("\nLIST\t=", list_1)

#CHANGE ELEMENT
list_1.append("Litch")
print("\nLIST AFTER APPENDING\t=", list_1)

list_1.pop(3)
print("\nLIST AFTER POPPING(3)\t=", list_1)

list_1.insert(3,"Pineapple")
print("\nLIST AFTER INSERTING(Pineapple)\t=", list_1)

list_1.remove("Pineapple")
print("\nLIST AFTER REMOVING(Pineapple)\t=", list_1)

del list_1[0]
print("\nLIST AFTER DELETING(0)\t=", list_1)

list_1[3:5] = ["Apple"]
print("\nLIST AFTER CHANGING \t=", list_1)

#IF it is not in range then add at last
list_1[5:6] = "BANANA"
print("\nLIST AFTER CHANGING \t=", list_1)

#CONVERTING LIST INTO TUPLE
tuple_1 = tuple(list_1)
print("\nFINAL TUPLE\t\t=", tuple_1)

# # we can delete tuple
del tuple_1
print("\nAFTER DELETING TUPLE\t\t=", tuple_1)
```

```
TUPLE - CHANGING, ADDING, DELETING ELEMENT USING LIST

TUPLE   = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawberry',
'Pomegranate')

LIST    = ['Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawberry',
'Pomegranate']

LIST AFTER APPENDING    = ['Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon',
'Strawberry', 'Pomegranate', 'Litch']

LIST AFTER POPPING(3)   = ['Mango', 'Cherry', 'Berry', 'Watermelon', 'Strawbe
rry', 'Pomegranate', 'Litch']

LIST AFTER INSERTING(Pineapple) = ['Mango', 'Cherry', 'Berry', 'Pineapple',
'Watermelon', 'Strawberry', 'Pomegranate', 'Litch']

LIST AFTER REMOVING(Pineapple)  = ['Mango', 'Cherry', 'Berry', 'Watermelon',
'Strawberry', 'Pomegranate', 'Litch']

LIST AFTER DELETING(0)  = ['Cherry', 'Berry', 'Watermelon', 'Strawberry', 'Po
megranate', 'Litch']

LIST AFTER CHANGING     = ['Cherry', 'Berry', 'Watermelon', 'Apple', 'Litch']

LIST AFTER CHANGING     = ['Cherry', 'Berry', 'Watermelon', 'Apple', 'Litch',
'B', 'A', 'N', 'A', 'N', 'A']

FINAL TUPLE             = ('Cherry', 'Berry', 'Watermelon', 'Apple', 'Litch',
'B', 'A', 'N', 'A', 'N', 'A')

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[9], line 39
     37 # # we can delete tuple
     38 del tuple_1
---> 39 print("\nAFTER DELETING TUPLE\t\t=", tuple_1)

NameError: name 'tuple_1' is not defined
```

In [10]:
```
print("TUPLE - ADDING TWO TUPLE ")

tuple_1 = ("Mango", "Cherry", "Berry", "Apple", "Watermelon", "Strawberry")
tuple_2 = ("Pomegranate",)
print("\nTUPLE 1\t\t\t=", tuple_1)
print("\nTUPLE 2\t\t\t=", tuple_2)
tuple_1 += tuple_2

print("\nTUPLE after adding\t=", tuple_1)
```

```
TUPLE - ADDING TWO TUPLE

TUPLE 1                 = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon',
'Strawberry')

TUPLE 2                 = ('Pomegranate',)

TUPLE after adding      = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon',
'Strawberry', 'Pomegranate')
```

In [11]:
```
print("TUPLE - UNPACKING TUPLE")

tuple_1 = ("Mango", "Cherry", "Berry")
varA, varB, varC = tuple_1
print("\nTUPLE 1\t\t=", tuple_1)
print("varA\t\t=" , varA)
print("varB\t\t=" , varB)
print("varC\t\t=" , varC)

tuple_2 = ("Mango", "Cherry", "Berry")
varA, varB, *varC = tuple_2
print("\nTUPLE 2\t\t=", tuple_2)
print("varA\t\t=" , varA)
print("varB\t\t=" , varB)
print("*varC\t\t=" , varC)
print(type(varB))
print(type(varC))


tuple_3 = ("Mango", "Cherry", "Berry" ,'Apple', 'Watermelon', 'Strawberry')
varA, varB, *varC = tuple_3
print("\nTUPLE 3\t\t=", tuple_3)
print("varA\t\t=" , varA)
print("varB\t\t=" , varB)
print("*varC\t\t=" , varC)
print(type(varB))
print(type(varC))


tuple_4= ("Mango", "Cherry", "Berry" ,'Apple', 'Watermelon', 'Strawberry')
varA, *varB, varC = tuple_4
print("\nTUPLE 4\t\t=", tuple_4)
print("varA\t\t=" , varA)
print("*varB\t\t=" , varB)
print("varC\t\t=" , varC)
print(type(varB))
print(type(varC))

tuple_5= ("Mango", "Cherry", "Berry" ,'Apple', 'Watermelon', 'Strawberry')
*varA, varB, varC = tuple_5
print("\nTUPLE 5\t\t=", tuple_5)
print("*varA\t\t=" , varA)
print("varB\t\t=" , varB)
print("varC\t\t=" , varC)
print(type(varA))
print(type(varC))
```

```
TUPLE - UNPACKING TUPLE

TUPLE 1         = ('Mango', 'Cherry', 'Berry')
varA            = Mango
varB            = Cherry
varC            = Berry

TUPLE 2         = ('Mango', 'Cherry', 'Berry')
varA            = Mango
varB            = Cherry
*varC           = ['Berry']
<class 'str'>
<class 'list'>

TUPLE 3         = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawb
erry')
varA            = Mango
varB            = Cherry
*varC           = ['Berry', 'Apple', 'Watermelon', 'Strawberry']
<class 'str'>
<class 'list'>

TUPLE 4         = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawb
erry')
varA            = Mango
*varB           = ['Cherry', 'Berry', 'Apple', 'Watermelon']
varC            = Strawberry
<class 'list'>
<class 'str'>

TUPLE 5         = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawb
erry')
*varA           = ['Mango', 'Cherry', 'Berry', 'Apple']
varB            = Watermelon
varC            = Strawberry
<class 'list'>
<class 'str'>
```

In [12]:

```
print("TUPLE - FOR_ IN LOOP")

tuple_1 = ("Mango", "Cherry", "Berry",'Apple', 'Watermelon', 'Strawberry' )
print("\nTUPLE 1 =", tuple_1)

for item in tuple_1:
    print("ELEMENT =" , item)
```

```
TUPLE - FOR_ IN LOOP

TUPLE 1 = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawberry')
ELEMENT = Mango
ELEMENT = Cherry
ELEMENT = Berry
ELEMENT = Apple
ELEMENT = Watermelon
ELEMENT = Strawberry
```

In [13]:
```
print("TUPLE - FOR_ IN LOOP (RANGE)")

tuple_1 = ("Mango", "Cherry", "Berry",'Apple', 'Watermelon', 'Strawberry' )
print("\nTUPLE 1 =", tuple_1)

for item in range(len(tuple_1)):
    print("ELEMENT =" , tuple_1[item])
```

```
TUPLE - FOR_ IN LOOP (RANGE)

TUPLE 1 = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawberry')
ELEMENT = Mango
ELEMENT = Cherry
ELEMENT = Berry
ELEMENT = Apple
ELEMENT = Watermelon
ELEMENT = Strawberry
```

In [14]:
```
print("TUPLE - MULTIPLY")

tuple_1 = ("Mango", "Cherry", "Berry",'Apple', 'Watermelon', 'Strawberry' )
print("\nTUPLE 1 =", tuple_1)
tuple_2 = tuple_1 * 6
print("\nTUPLE 2 =", tuple_2)
```

```
TUPLE - MULTIPLY

TUPLE 1 = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawberry')

TUPLE 2 = ('Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawberry',
'Mango', 'Cherry', 'Berry', 'Apple', 'Watermelon', 'Strawberry', 'Mango', 'Ch
erry', 'Berry', 'Apple', 'Watermelon', 'Strawberry', 'Mango', 'Cherry', 'Berr
y', 'Apple', 'Watermelon', 'Strawberry', 'Mango', 'Cherry', 'Berry', 'Apple',
'Watermelon', 'Strawberry', 'Mango', 'Cherry', 'Berry', 'Apple', 'Watermelo
n', 'Strawberry')
```