Assignment 6 - Python

Tuple Data Structure

In [1]:
```
tup1 = ()   # Empty Tuple
print(tup1)
```

```
()
```

In [2]:
```
tup2 = (10,30,60)         # Tuple of integer numbers
print(tup2)
```

```
(10, 30, 60)
```

In [3]:
```
tup3 = (10.77,30.56,60.89)         # Tuple of float numbers
print(tup3)
```

```
(10.77, 30.56, 60.89)
```

In [13]:
```
tup4 = ("one","two","three")            # Tuple of string
print(tup4)
```

```
('one', 'two', 'three')
```

In [5]:
```
tup5 = ('Vihari', 25, (50,100), (150,90))     # Nested Tuples
print(tup5)
```

```
('Vihari', 25, (50, 100), (150, 90))
```

In [6]:
```
tup6 = (100, 'Vihari', 17.765)          # Tuple of mixed data types
print(tup6)
```

```
(100, 'Vihari', 17.765)
```

In [7]:
```
tup7 = ('Vihari', 25, [50,100], [150,90],{'John', 'David'}, (99,22,33))
print(tup7)
```

```
('Vihari', 25, [50, 100], [150, 90], {'David', 'John'}, (99, 22, 33))
```

In [8]:
```
print(len(tup7))   # Length of Tuple
```

```
6
```

Tuple Indexing

In [9]:
```
print(tup2[0])
```

```
10
```

In [10]:
```
print(tup4[0])
```

```
one
```

In [11]:
```
print(tup4[0][0])
```

```
o
```

In [14]:
```
print(tup4[-1])
```

```
three
```

In [15]:
```
print(tup5[-1])
```

```
(150, 90)
```

Tuple Slicing

In [16]:
```
mytuple = ('one','two','three','four','five','six','seven','eight')
print(mytuple)
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [17]:
```
print(mytuple[0:3])
```

```
('one', 'two', 'three')
```

In [18]:
```
print(mytuple[2:5])
```

```
('three', 'four', 'five')
```

In [19]:
```
print(mytuple[:3])
```

```
('one', 'two', 'three')
```

In [20]:
```
print(mytuple[:2])    # Returns first two items
```

```
('one', 'two')
```

In [21]:
```
print(mytuple[-3:])    # Returns last three items
```

```
('six', 'seven', 'eight')
```

In [22]:
```
print(mytuple[-2:])
```

```
('seven', 'eight')
```

In [23]:
```
print(mytuple[-1:])
```

```
('eight',)
```

In [24]:
```
print(mytuple[:])
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

Remove & Change Items

In [25]:
```
mytuple = ('one','two','three','four','five','six','seven','eight')
print(mytuple)
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [27]:
```
del mytuple[0]     # Tuples are immutable which means we can't Delete tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 del mytuple[0]     # Tuples are immutable which means we can't Delete tuple
items

TypeError: 'tuple' object doesn't support item deletion
```

In [28]:
```
mytuple[0] = 1   # Tuples are immutable which means we can't Delete tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 1
----> 1 mytuple[0] = 1

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

In [29]:
```
del mytuple      # Deleting entire tuple object is possible
```

Loop through a Tuple

In [30]:
```
mytuple = ('one','two','three','four','five','six','seven','eight')
print(mytuple)
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [31]:
```
for i in mytuple:
    print(i)
```

```
one
two
three
four
five
six
seven
eight
```

In [32]:
```
for i in enumerate(mytuple):
    print(i)
```

```
(0, 'one')
(1, 'two')
(2, 'three')
(3, 'four')
(4, 'five')
(5, 'six')
(6, 'seven')
(7, 'eight')
```

Tuple Membership

In [33]:
```
mytuple = ('one','two','three','four','five','six','seven','eight')
print(mytuple)
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [34]:
```
'one' in mytuple       # check if 'one' exists in the tuple
```

Out[34]: True

In [35]:
```
'ten' in mytuple       # check if 'ten' exists in the tuple
```

Out[35]: False

In [37]:
```
if 'three' in mytuple:          # check if 'three' exist in the tuple
    print("Three is available in the Tuple")
else:
    print("Three is not available in the Tuple")
```

```
Three is available in the Tuple
```

In [38]:
```
if 'eleven' in mytuple:          # check if 'eleven' exist in the tuple
    print("Eleven is available in the Tuple")
else:
    print("Eleven is not available in the Tuple")
```

```
Eleven is not available in the Tuple
```

Index Position

In [39]:
```
mytuple = ('one','two','three','four','five','six','seven','eight')
print(mytuple)
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [40]:
```
mytuple.index('one')         # Index of first element is equal to 'one'
```

Out[40]: 0

In [41]:
```
mytuple.index('five')         # Index of first element is equal to 'five'
```

Out[41]: 4

In [42]:
```
mytuple1 = ('one','two','three','four','one','one','two','three')
print(mytuple)
```

```
('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

In [43]:
```
mytuple1.index('one')       # Index of first element is equal to 'one'
```

Out[43]: 0

Sorting

In [44]:
```
mytuple2 = (43,67,99,12,6,90,67)
print(mytuple2)
```

```
(43, 67, 99, 12, 6, 90, 67)
```

In [46]:
```
sorted(mytuple2)            # Returns a new sorted tuple and doesn't change origina
```

Out[46]:
```
[6, 12, 43, 67, 67, 90, 99]
```

In [47]:
```
sorted(mytuple2, reverse=True)     # Sort in Descending Order)
```

Out[47]:
```
[99, 90, 67, 67, 43, 12, 6]
```