Tuples

1. Tuple is similar to List except that the objects in tuple are immutable which means we cannot change the elements of a tuple once assigned.
2. When we do not want to change the data over time, tuple is a preferred data type.
3. Iterating over the elements of a tuple is faster compared to iterating over a list.

Tuple Creation

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

```
In [2]: tup2 = (10,30,60)  # tuple of integers numbers
```

```
In [3]: tup3 = (10.77,30.66,60.89) # tuple of float numbers
```

```
In [4]: tup4 = ('one','two' , "three") # tuple of strings
```

```
In [9]: tup5 = ('Arya', 25 ,(50, 100),(150, 90)) # Nested tuples
```

```
In [8]: tup6 = (100, 'Arya', 17.765) # Tuple of mixed data types
```

```
In [10]: tup7 = ('Arya', 25 ,[50, 100],[150, 90] , {'John' , 'David'} , (99,22,33))
```

```
In [11]: len(tup7) #Length of list
```

```
Out[11]: 6
```

Tuple Indexing

```
In [12]: tup2[0] # Retreive first element of the tuple
```

```
Out[12]: 10
```

```
In [13]: tup4[0] # Retreive first element of the tuple
```

```
Out[13]: 'one'
```

```
In [14]: tup4[0][0] # Nested indexing - Access the first character of the first tuple e
```

```
Out[14]: 'o'
```

```
In [15]: tup4[-1] # Last item of the tuple

Out[15]: 'three'

In [16]: tup5[-1] # Last item of the tuple

Out[16]: (150, 90)
```

Tuple Slicing

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

In [19]: mytuple[0:3] # Return all items from 0th to 3rd index location excluding the i

Out[19]: ('one', 'two', 'three')

In [20]: mytuple[2:5] # List all items from 2nd to 5th index location excluding the ite

Out[20]: ('three', 'four', 'five')

In [21]: mytuple[:3] # Return first three items

Out[21]: ('one', 'two', 'three')

In [22]: mytuple[:2] # Return first two items

Out[22]: ('one', 'two')

In [23]: mytuple[-3:] # Return last three items

Out[23]: ('six', 'seven', 'eight')

In [24]: mytuple[-2:] # Return last two items

Out[24]: ('seven', 'eight')

In [25]: mytuple[-1] # Return last item of the tuple

Out[25]: 'eight'

In [26]: mytuple[:] # Return whole tuple

Out[26]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

Remove & Change Items

```
In [34]: mytuple
```

```
Out[34]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

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

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

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

```
In [29]: mytuple[0] = 1 # Tuples are immutable which means we can't CHANGE tuple items
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[29], line 1
----> 1 mytuple[0] = 1

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

Loop through a tuple

```
In [35]: mytuple
```

```
Out[35]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [37]: for i in mytuple:
             print(i)
```

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

```
In [38]: for i in enumerate(mytuple):
             print(i)
```

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

Count

```
In [39]: mytuple1 =('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

```
In [40]: mytuple1.count('one') # Number of times item "one" occurred in the tuple.

Out[40]: 3
```

```
In [41]: mytuple1.count('two') # Occurence of item 'two' in the tuple

Out[41]: 2
```

```
In [42]: mytuple1.count('four') #Occurence of item 'four' in the tuple

Out[42]: 1
```

Tuple Membership

```
In [43]: mytuple

Out[43]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [44]: 'one' in mytuple # Check if 'one' exist in the list

Out[44]: True
```

```
In [45]: 'ten' in mytuple # Check if 'ten' exist in the list

Out[45]: False
```

```
In [46]: if 'three' in mytuple: # Check if 'three' exist in the list
             print('Three is present in the tuple')
         else:
             print('Three is not present in the tuple')

Three is present in the tuple
```

```
In [47]: if 'eleven' in mytuple: # Check if 'eleven' exist in the list
             print('eleven is present in the tuple')
```

```
else:
    print('eleven is not present in the tuple')
```

```
eleven is not present in the tuple
```

Index Position

```
In [48]: mytuple
```

```
Out[48]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight')
```

```
In [49]: mytuple.index('one') # Index of first element equal to 'one'
```

```
Out[49]: 0
```

```
In [50]: mytuple.index('five') # Index of first element equal to 'five'
```

```
Out[50]: 4
```

```
In [51]: mytuple1
```

```
Out[51]: ('one', 'two', 'three', 'four', 'one', 'one', 'two', 'three')
```

```
In [52]: mytuple1.index('one') # Index of first element equal to 'one'
```

```
Out[52]: 0
```

Sorting

```
In [53]: mytuple2 = (43,67,99,12,6,90,67)
```

```
In [54]: sorted(mytuple2) # Returns a new sorted list and doesn't change original tuple
```

```
Out[54]: [6, 12, 43, 67, 67, 90, 99]
```

```
In [55]: sorted(mytuple2, reverse=True) # Sort in descending order
```

```
Out[55]: [99, 90, 67, 67, 43, 12, 6]
```

Tuple is completed