

```
tup1 = () # Empty tuple
tup2 = (10,30,60) # tuple of integers numbers
tup3 = (10.77,30.66,60.89) # tuple of float numbers
tup4 = ('one','two' , "three") # tuple of strings
tup5 = ('Asif', 25 ,(50, 100),(150, 90)) # Nested tuples
tup6 = (100, 'Asif', 17.765) # Tuple of mixed data types
tup7 = ('Asif', 25 ,[50, 100],[150, 90] , {'John' , 'David'} , (99,22,33))
```

```
tup1 ,tup2,tup3,tup4
```

```
⇒ ((), (10, 30, 60), (10.77, 30.66, 60.89), ('one', 'two', 'three'))
```

```
tup5
```

```
⇒ ('Asif', 25, (50, 100), (150, 90))
```

```
tup6 ,tup7
```

```
⇒ ((100, 'Asif', 17.765),
    ('Asif', 25, [50, 100], [150, 90], {'David', 'John'}, (99, 22, 33)))
```

```
len(tup7) #Length of list
```

```
⇒ 6
```

```
tup2[0] # Retrieve first element of the tuple
```

```
⇒ 10
```

```
tup4[0][0] # Nested indexing - Access the first character of the first tuple ele
```

```
⇒ 'o'
◀────────────────────────────────────────────────────────────────────────────────▶
```

```
tup5[-1] # Last item of the tuple
```

```
⇒ (150, 90)
```

```
tup4[0] # Retrieve first element of the tuple
```

```
⇒ 'one'
◀────────────────────────────────────────────────────────────────────────────────▶
```

```
tup4[-1] # Last item of the tup
```

```
⇒ 'three'
◀────────────────────────────────────────────────────────────────────────────────▶
```

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

```
mytuple[0:3]
```

```
➞ ('one', 'two', 'three')
```

```
mytuple[2:5] # List all items from 2nd to 5th index location excluding the item
```

```
➞ ('three', 'four', 'five')
```

```
mytuple[:3] # Return first three items
```

```
➞ ('one', 'two', 'three')
```

```
mytuple[:2] # Return first two items
```

```
➞ ('one', 'two')
```

```
mytuple[-3:] # Return last three items
```

```
➞ ('six', 'seven', 'eight')
```

```
mytuple[-2:] # Return last two items
```

```
➞ ('seven', 'eight')
```

```
mytuple[-1] # Return last item of the tuple
```

```
➞ 'eight'
```

```
mytuple[:]
```

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

```
mytuple
```

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

```
mytuple[0] = 1
```

```
➞ -----  
TypeError                                Traceback (most recent call last)  
  <ipython-input-26-4cf492702bfd> in <cell line: 0>()  
    ----> 1 mytuple[0] = 1  
  
TypeError: 'tuple' object does not support item assignment
```

Next steps: [Explain error](#)

```
del mytuple[0]
```

```
➞ -----  
TypeError                                Traceback (most recent call last)  
  <ipython-input-25-667a276aa503> in <cell line: 0>()  
    ----> 1 del mytuple[0]  
  
TypeError: 'tuple' object doesn't support item deletion
```

Next steps: [Explain error](#)

```
mytuple=('one','two','three'+'four')
```

```
mytuple.index('one')
```

```
⇒ 0
```

Start coding or [generate](#) with AI.

```
mytuple
```

```
⇒ ('one', 'two', 'threefour')
```

```
for i in mytuple:  
    print(i)
```

```
⇒ one  
    two  
    threefour
```

```
mytuple2 = (43,67,99,12,6,90,67)
```

```
sorted(mytuple2) # display orted order
```

```
⇒ [6, 12, 43, 67, 67, 90, 99]
```

```
mytuple2
```

```
⇒ (43, 67, 99, 12, 6, 90, 67)
```

```
sorted(mytuple2, reverse=True)
```

```
⇒ [99, 90, 67, 67, 43, 12, 6]
```

```
mytuple2.count(48)
```

```
⇒ 0
```

```
for i in enumerate(mytuple):  
    print(i)
```

```
⇒ (0, 'one')  
    (1, 'two')  
    (2, 'threefour')
```

```
'one' in mytuple
```

```
⇒ True
```