

Tuple creation

```
tup1 = () #Empty tuple

tup2 =(10,20,30)      #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','Dravid'},(99,22,33))
```

Double-click (or enter) to edit

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

```
↵ 6
```

```
len(tup3)
```

```
↵ 3
```

Tuple indexing

```
tup2[0]
```

```
↵ 10
```

```
tup4[0]
```

```
↵ 'one'
```

```
tup4[0][0]
```

```
↵ '^'
```

```
tup4[-1]
```

```
↵ 'three'
```

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

Tuple Slicing

```
mytuple =( 'one','two','three','four','five','six','seven','eight','nine','ten')
mytuple[0:3]
```

```
↵ ('one', 'two', 'three')
```

```
mytuple[2:5]
```

```
↵ ('three', 'four', 'five')
```

```
mytuple[:3]
```

```
➦ ('one', 'two', 'three')
```

```
mytuple[:2]
```

```
➦ ('one', 'two')
```

```
mytuple[-3:]
```

```
➦ ('eight', 'nine', 'ten')
```

```
mytuple[-2:]
```

```
➦ ('nine', 'ten')
```

```
mytuple[-1:]
```

```
➦ ('ten',)
```

```
mytuple[:]
```

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

Remove & change items

```
mytuple
```

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

```
del mytuple[0]
```

```
➦ -----
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-29-1004790074.py in <cell line: 0>()
----> 1 del mytuple[0]

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

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

```
➦ -----
TypeError                                 Traceback (most recent call last)
/tmp/ipython-input-30-3796157582.py in <cell line: 0>()
----> 1 mytuple[0] =1

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

```
del mytuple
```

Loop through a tuple

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

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

```
➦ one
two
three
four
five
six
seven
eight
nine
ten
```

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

```

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

Tuple Membership

```
mytuple
```

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

```
'one' in mytuple
```

```
True
```

```
'ten' in mytuple
```

```
True
```

```
if 'three' in mytuple:
    print('Three is present in the tuple')
else:
    print('Three is not present in the tuple')
```

```
Three is present in the tuple
```

```
if 'eleven' in mytuple:
    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

```
mytuple
```

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

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

```
0
```

```
mytuple.index('ten')
```

```
9
```

```
mytuple.index('nine')
```


```
8
```

Sorting

```
mytuple2 =(40,43,67,90,56,98,100,45)
sorted(mytuple2)
```

```
[40, 43, 45, 56, 67, 90, 98, 100]
```

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

 [100, 98, 90, 67, 56, 45, 43, 40]