

```

1 # Tuple
2
3 """
4 TUPLE:
5     --> An ordered collection or grouping of items!.
6     --> var_name = (a, b, c, d).
7     --> Tuple is Immutable - can NEVER be changed!.
8     --> Tuples are faster than lists.
9     --> It makes your code safer.
10    --> Valid Keys in a dictionary.
11    --> Some methods return them to you - like .items() method when working
with dictionaries.
12    --> It's Best When Using as Enumeration.
13    --> Can Use Slices Like Lists.
14 """
15
16 """
17 Creating Tuple:
18     --> Two Ways:
19         --> () syntax.
20         --> tuple() function
21 """
22 x = (1, 2, 3)
23 print(x)
24
25 y = tuple(['Abhishek', 'Dylan'])
26 print(y)
27
28 # Access Data
29 print(x[0])
30
31 # x[1] = 6 - It Fails Because tuple does not support item assignment
(Immutable in a nutshell)
32
33 months = ('January', 'February', 'March', 'April', 'May')
34 print(months)
35
36 # Tuples can be used as keys in dictionaries
37 locations = {
38     (35.6895, 39.6917): "Tokyo Office",
39     (40.7128, 74.0060): "New York Office",
40     (37.7749, 122.4194): "San Francisco Office"
41 }
42
43 print(locations[(35.6895, 39.6917)])
44
45 """
46 Iterating Over Tuples:
47     --> can use for loop to iterate over tuples.
48 """
49 for month in months:
50     print(month)
51
52 """
53 Tuples Built-In Methods:
54     --> 1. count(x) -> Returns Number of occurrences of x in the targeted
tuple.
55     --> 2. index(x) -> Return Index First Occurrence of given value(x).
56 """
57 z = (1, 2, 3, 3, 3, 5, 5, 6, 6, 6, 6)

```

```
58 print(z.count(1))
59 print(z.count(3))
60 print(z.count(6))
61
62 print(z.index(6))
63
64 """
65 Nested Tuples
66 """
67 nested_tuple = (1, 2, 3, ('Programming', 'Music'))
68 print(nested_tuple)
69
```