**Tuples in Python**

* Difficulty Level : [Easy](https://www.geeksforgeeks.org/easy/)
* Last Updated : 19 Jul, 2022

**Python Tuple** is a collection of objects separated by commas. In some ways, a tuple is similar to a list in terms of indexing, nested objects, and repetition but a tuple is immutable, unlike lists which are mutable.

**Creating Tuples in Python**

To create a tuple we will use () operators.

* Python

|  |
| --- |
| var = ("Geeks", "for", "Geeks")  print(var) |

**Output:**

('Geeks', 'for', 'Geeks')

*Note: In case your generating a tuple with a single element, make sure to add a comma after the element.*

**Accessing Values in Tuples in Python**

**Method 1: Using Positive Index**

Using square brackets we can get the values from tuples in [Python](https://www.geeksforgeeks.org/python-programming-language/).

* Python3

|  |
| --- |
| var = ("Geeks", "for", "Geeks")    print("Value in Var[0] = ", var[0])  print("Value in Var[1] = ", var[1])  print("Value in Var[2] = ", var[2]) |

**Output:**

Value in Var[0] = Geeks

Value in Var[1] = for

Value in Var[2] = Geeks

**Method 2: Using Negative Index.**

In the above methods, we use the positive index to access the value in Python, and here we will use -ve index within [].

* Python3

|  |
| --- |
| var = ("Geeks", "for", "Geeks")    print("Value in Var[-3] = ", var[-3])  print("Value in Var[-2] = ", var[-2])  print("Value in Var[-1] = ", var[-1]) |

**Concatenation of Tuples in Python**

To concatenate the Python tuple we will use plus operators(+).

* Python

|  |
| --- |
| # Code for concatenating 2 tuples    tuple1 = (0, 1, 2, 3)  tuple2 = ('python', 'geek')    # Concatenating above two  print(tuple1 + tuple2) |

**Output:**

(0, 1, 2, 3, 'python', 'geek')

**Nesting of Tuples in Python**

* Python

|  |
| --- |
| # Code for creating nested tuples    tuple1 = (0, 1, 2, 3)  tuple2 = ('python', 'geek')  tuple3 = (tuple1, tuple2)  print(tuple3) |

**Output :**

((0, 1, 2, 3), ('python', 'geek'))

**Repetition Tuples in Python**

* Python

|  |
| --- |
| # Code to create a tuple with repetition    tuple3 = ('python',)\*3  print(tuple3) |

**Output:**

('python', 'python', 'python')

Try the above without a comma and check. You will get tuple3 as a string ‘pythonpythonpython’.

**Immutable Tuples in Python**

* Python

|  |
| --- |
| # code to test that tuples are immutable    tuple1 = (0, 1, 2, 3)  tuple1[0] = 4  print(tuple1) |

**Output:**

Traceback (most recent call last):

File "e0eaddff843a8695575daec34506f126.py", line 3, in

tuple1[0]=4

TypeError: 'tuple' object does not support item assignment

**Slicing Tuples in Python**

* Python

|  |
| --- |
| # code to test slicing    tuple1 = (0 ,1, 2, 3)  print(tuple1[1:])  print(tuple1[::-1])  print(tuple1[2:4]) |

**Output:**

(1, 2, 3)

(3, 2, 1, 0)

(2, 3)

**Deleting a Tuple**

* Python

|  |
| --- |
| # Code for deleting a tuple    tuple3 = ( 0, 1)  del tuple3  print(tuple3) |

**Error:**

Traceback (most recent call last):

File "d92694727db1dc9118a5250bf04dafbd.py", line 6, in <module>

print(tuple3)

NameError: name 'tuple3' is not defined

**Output:**

(0, 1)

**Finding Length of a Tuple**

* Python

|  |
| --- |
| # Code for printing the length of a tuple    tuple2 = ('python', 'geek')  print(len(tuple2)) |

**Output:**

2

**Converting list to a Tuple**

* Python

|  |
| --- |
| # Code for converting a list and a string into a tuple    list1 = [0, 1, 2]  print(tuple(list1))  print(tuple('python')) # string 'python' |

**Output:**

(0, 1, 2)

('p', 'y', 't', 'h', 'o', 'n')

Takes a single parameter which may be a list, string, set or even a dictionary( only keys are taken as elements) and converts them to a tuple.

**Tuples in a loop**

* Python

|  |
| --- |
| # python code for creating tuples in a loop    tup = ('geek',)  n = 5  # Number of time loop runs  for i in range(int(n)):      tup = (tup,)      print(tup) |

**Output:**

(('geek',),)

((('geek',),),)

(((('geek',),),),)

((((('geek',),),),),)

(((((('geek',),),),),),)