**Tuple**

Tuples are used to store multiple items in a single variable.

Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are [List](https://www.w3schools.com/python/python_lists.asp), [Set](https://www.w3schools.com/python/python_sets.asp), and [Dictionary](https://www.w3schools.com/python/python_dictionaries.asp), all with different qualities and usage.

A tuple is a collection which is ordered and unchangeable.

Tuples are written with round brackets.

Ex:-

thistuple = ("apple", "banana", "cherry")  
print(thistuple)

## Tuple Items

Tuple items are ordered, unchangeable, and allow duplicate values.

**Ordered**

When we say that tuples are ordered, it means that the items have a defined order, and that order will not change.

**Unchangeable**

Tuples are unchangeable, meaning that we cannot change, add or remove items after the tuple has been created.

**Allow Duplicates**

Since tuple are indexed, tuples can have items with the same value:

Ex:-

thistuple = ("apple", "banana", "cherry", "apple", "cherry")  
print(thistuple)

Ex:

thistuple = ("apple", "banana", "cherry")  
print(len(thistuple))

Output:

('apple', 'banana', 'cherry')

Ex1:

thistuple = ("apple", "banana", "cherry", "apple", "cherry")  
print(thistuple)

output:

('apple', 'banana', 'cherry', 'apple', 'cherry')

**Len():**

**To determine how many items a tuple has, use the len() function  
Ex:**thistuple = ("apple", "banana", "cherry")  
print(len(thistuple))

Output:

3

**Data types:**

* Tuple items can be of any data type:
* String, int and boolean data types:  
    
  tuple1 = ("apple", "banana", "cherry")  
  tuple2 = (1, 5, 7, 9, 3)  
  tuple3 = (True, False, False)  
  print(tuple1)  
  print(tuple2)  
  print(tuple3)

Output:

('apple', 'banana', 'cherry')

(1, 5, 7, 9, 3)

(True, False, False)

**Constructor:**

* It is also possible to use the tuple() constructor to make a tuple.

thistuple = tuple(("apple", "banana", "cherry")) # note the double round-brackets  
print(thistuple)

**Output**:

('apple', 'banana', 'cherry')

## Python Collections (Arrays)

There are four collection data types in the Python programming language:

* [List](https://www.w3schools.com/python/python_lists.asp) is a collection which is ordered and changeable. Allows duplicate members.
* **Tuple** is a collection which is ordered and unchangeable. Allows duplicate members.
* [Set](https://www.w3schools.com/python/python_sets.asp) is a collection which is unordered and unindexed. No duplicate members.
* [Dictionary](https://www.w3schools.com/python/python_dictionaries.asp) is a collection which is ordered\* and changeable. No duplicate members.

\*As of Python version 3.7, dictionaries are *ordered*. In Python 3.6 and earlier, dictionaries are *unordered*.

When choosing a collection type, it is useful to understand the properties of that type. Choosing the right type for a particular data set could mean retention of meaning, and, it could mean an increase in efficiency or security.

**Accesing index:**

* You can access tuple items by referring to the index number, inside square brackets  
  Ex:-

thistuple = ("apple", "banana", "cherry")  
print(thistuple[1])

**Output:**

**banana**

**Negitive Index**:

Ex:-  
thistuple = ("apple", "banana", "cherry")  
print("negitive index:",thistuple[-1])

**Output:**

negitive index: cherry  
**range of index :-**

**Ex:**thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
print("range:",thistuple[2:5])

**output:**

**range: ('cherry', 'orange', 'kiwi')**  
  
**By leaving out the end value, the range will go on to the end of the list:**

Ex:

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
print("First to end value:",thistuple[:4])

**output:**

**First to end value: ('apple', 'banana', 'cherry', 'orange')  
  
start index to end value of index:**

**Ex:**

thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
print("start to end:",thistuple[2:])

**Output:**

**start to end: ('cherry', 'orange', 'kiwi', 'melon', 'mango')**

**This example returns the items from index -4 (included) to index -1 (excluded)**

Ex:  
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")  
print(thistuple[-4:-1])

**Output:**

**('orange', 'kiwi', 'melon')**

**Check if "apple" is present in the tuple:**

Ex:  
  
thistuple = ("apple", "banana", "cherry")  
if "apple" in thistuple:  
 print("Yes, 'apple' is in the fruits tuple")

**Output:**

**Yes, 'apple' is in the fruits tuple.**

## Change Tuple Values

Once a tuple is created, you cannot change its values. Tuples are **unchangeable**, or **immutable** as it also is called.

But there is a workaround. You can convert the tuple into a list, change the list, and convert the list back into a tuple.

**Convert the tuple into a list to be able to change it:**Ex:  
x = ("apple", "banana", "cherry")  
y = list(x)  
y[1] = "kiwi"  
x = tuple(y)  
print(x)

**Output:**

**('apple', 'kiwi', 'cherry')**

**You cannot add items to a tuple:**Ex:-  
thistuple = ("apple", "banana", "cherry")  
thistuple.append("orange") # This will raise an error  
print(thistuple)

**Output:**

**Traceback (most recent call last):**

**File "C:/Users/sekha/PycharmProjects/tuple1/tuple8(add).py", line 4, in <module>**

**thistuple.append("orange") # This will raise an error**

**AttributeError: 'tuple' object has no attribute 'append'**

**Convert the tuple into a list, add "orange", and convert it back into a tuple**Ex:-  
thistuple = ("apple", "banana", "cherry")  
y = list(thistuple)  
y.append("orange")  
thistuple = tuple(y)  
print(thistuple)

**Output:**

**('apple', 'banana', 'cherry', 'orange')**

## Remove Items

**Note:** You cannot remove items in a tuple.

Tuples are **unchangeable**, so you cannot remove items from it, but you can use the same workaround as we used for changing and adding tuple items:

**Convert the tuple into a list, remove "apple", and convert it back into a tuple:**

Ex:-  
thistuple = ("apple", "banana", "cherry")  
y = list(thistuple)  
y.remove("apple")  
thistuple = tuple(y)  
  
print(thistuple)

**Output:**

**('banana', 'cherry')**

**The del keyword can delete the tuple completely:**  
Ex:-  
thistuple = ("apple", "banana", "cherry")  
del thistuple  
print(thistuple) #this will raise an error because the tuple no longer exists

Output:

Traceback (most recent call last):

File "C:/Users/sekha/PycharmProjects/tuple1/tuple11(del).py", line 5, in <module>

print(thistuple) #this will raise an error because the tuple no longer exists

NameError: name 'thistuple' is not defined

**Unpacking a tuple:**  
  
fruits = ("apple", "banana", "cherry")

(green, yellow, red) = fruits  
  
print(green)  
print(yellow)  
print(red)

**Output:**

**apple**

**banana**

**cherry**

## Using Asterisk\*

If the number of variables is less than the number of values, you can add an \* to the variable name and the values will be assigned to the variable as a list:

**Ex: using you can add an \* to the variable name and the values will be assigned to the**

fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")  
  
(green, yellow, \*red) = fruits

print(green)  
print(yellow)  
print(red)

output:

apple

banana

['cherry', 'strawberry', 'raspberry']

**Add a list of values the "tropic" variable:**  
Ex:  
fruits = ("apple", "mango", "papaya", "pineapple", "cherry")  
  
(green, \*tropic, red) = fruits # \*tropic to assign the remaing values.  
  
print(green)  
print(tropic)  
print(red)

**Output:**

**apple**

**['mango', 'papaya', 'pineapple']**

**cherry**

**for:**

Ex:

thistuple = ("apple", "banana", "cherry","ram")  
for x in thistuple:  
 print(x)

**output:**

**apple**

**banana**

**cherry**

**ram**

**Print all items by referring to their index number:**  
Ex:  
thistuple = ("apple", "bat", "camel")  
for i in range(len(thistuple)):  
 print(thistuple[i])

**Output:**

**apple**

**bat**

**camel**

**print all items, using a while loop to go through all the index numbers**:  
Ex:  
thistuple = ("computer", "suprem", "human")  
i = 0  
while i < len(thistuple):  
 print(thistuple[i])  
 i = i + 1

**Output:**

**computer**

**suprem**

**human**

**Join two tuples:**  
  
tuple1 = ("a", "b" , "c")  
tuple2 = (1, 2, 3)  
  
tuple3 = tuple1 + tuple2  
print(tuple3)

**Output:**

**('a', 'b', 'c', 1, 2, 3)**

**Multiply the fruits tuple by 2:**  
  
fruits = ("ram", "bad", "chandu")  
mytuple = fruits \* 2  
  
print(mytuple)

**Output:**

**('ram', 'bad', 'chandu', 'ram', 'bad', 'chandu')**

**Tuple-methods**

**Count():-**

## Definition and Usage:

## The count() method returns the number of times a specified value appears in the tuple.

## Syntax

***tuple*.count(*value*)**

## Parameter Values

|  |  |
| --- | --- |
| Parameter | Description |
| *value* | Required. The item to search for |

**Return the number of times the value 5 appears in the tuple:**

Ex:  
  
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)  
  
x = thistuple.count(5)  
  
print(x)

Output:

2

## Index():

## Definition and Usage

* The index() method finds the first occurrence of the specified value.
* The index() method raises an exception if the value is not found.

## Syntax

***tuple*.index(*value*)**

## Parameter Values

|  |  |
| --- | --- |
| Parameter | Description |
| *value* | Required. The item to search for |

**Search for the first occurrence of the value 8, and return its position:**  
  
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)  
  
x = thistuple.index(8)  
  
print(x)

Output:

3