Python Tuple

In [2]:
```
thistuple = ("apple", "banana", "cherry")
thistuple
```

Out[2]:
```
('apple', 'banana', 'cherry')
```

In [6]:
```
thistuple = ("b","d","c","a","e")
thistuple
```

Out[6]:
```
('b', 'd', 'c', 'a', 'e')
```

In [8]:
```
len(thistuple)
```

Out[8]:
```
5
```

Create Tuple With One Item

In [12]:
```
thistuple = ("apple")
type(thistuple)
```

Out[12]:
```
str
```

In [14]:
```
thistuple = ("apple",)
type(thistuple)
```

Out[14]:
```
tuple
```

Tuple items can be of any data type:

In [17]:
```
tuple1 = ("apple", "banana", "cherry")
tuple2 = (1, 5, 7, 9, 3)
tuple3 = (True, False, False)
```

In [19]:
```
print(tuple1)
print(tuple2)
print(tuple3)
```

```
('apple', 'banana', 'cherry')
(1, 5, 7, 9, 3)
(True, False, False)
```

In [21]:
```
type(True)
```

Out[21]: bool

In [23]:
```
print(type(tuple1))
```

```
<class 'tuple'>
```

The tuple() Constructor

In [28]:
```
# Using the tuple() method to make a tuple:

thistuple = tuple(("apple","Orange","Carrot"))
print(thistuple)
print(type(thistuple))
```

```
('apple', 'Orange', 'Carrot')
<class 'tuple'>
```

Access Tuple Items

In [31]:
```
thistuple = ("apple", "banana", "cherry")
thistuple
```

Out[31]: ('apple', 'banana', 'cherry')

In [33]:
```
thistuple[1]
```

Out[33]: 'banana'

In [37]:
```
thistuple[-1]
```

Out[37]: 'cherry'

In [39]:
```
thistuple[1][2]
```

Out[39]: 'n'

In [43]:
```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "man
thistuple
```

Out[43]: ('apple', 'banana', 'cherry', 'orange', 'kiwi', 'melon', 'mango')

In [45]:
```
thistuple[2:5]
```

Out[45]: ('cherry', 'orange', 'kiwi')

In [49]:
```
thistuple[-5:-1]
```

Out[49]: ('cherry', 'orange', 'kiwi', 'melon')

In [51]:
```
thistuple[-1:3]
```

Out[51]: ()

In [53]:
```
thistuple[-1:-3]
```

Out[53]: ()

Update Tuples

In [58]:
```
thistuple = ("apple","banana","orange","Beetroot")
print(thistuple)
type(thistuple)
```

```
('apple', 'banana', 'orange', 'Beetroot')
```

Out[58]: tuple

In [60]:
```
# update tuple to list and update the values
newtuple = list(thistuple)
print(newtuple)
print(type(newtuple))
```

```
['apple', 'banana', 'orange', 'Beetroot']
<class 'list'>
```

In [62]:
```
newtuple[0] = "Kiwi"
print(newtuple)
```

```
['Kiwi', 'banana', 'orange', 'Beetroot']
```

Add New item to Tuple

In [65]:
```
tuple1 = ("orange","grapes","dragon","berries")
print(tuple1)
print(type(tuple1))
```

```
('orange', 'grapes', 'dragon', 'berries')
<class 'tuple'>
```

In [67]:
```
newT = list(tuple1)
print(newT)
print(type(newT))
```

```
['orange', 'grapes', 'dragon', 'berries']
<class 'list'>
```

In [69]:
```
newT.append("Programante")
print(newT)
```

```
['orange', 'grapes', 'dragon', 'berries', 'Programante']
```

In [71]:
```
type(tuple1)
```

Out[71]:
```
tuple
```

In [73]:
```
type(newT)
```

Out[73]:
```
list
```

In [75]:
```
tuple1 = tuple(newT)
print(tuple1)
print(type(tuple1))
```

```
('orange', 'grapes', 'dragon', 'berries', 'Programante')
<class 'tuple'>
```

Add tuple to a tuple

In [78]:
```
tuple2 = ("dog","cat","pet","rose","plant")
tuple2
```

Out[78]:
```
('dog', 'cat', 'pet', 'rose', 'plant')
```

In [80]:
```
newT = ("apple",)
tuple2 = tuple2 + newT
print(tuple2)
```

```
('dog', 'cat', 'pet', 'rose', 'plant', 'apple')
```

In [82]:
```
tuple2 = newT + tuple2
```

In [84]:
```
print(tuple2)
```

```
('apple', 'dog', 'cat', 'pet', 'rose', 'plant', 'apple')
```

In [88]:
```
tuple3 = ("newItem",)
tuple2 = tuple3 + tuple2
print(tuple2)
```

```
('newItem', 'newItem', 'apple', 'dog', 'cat', 'pet', 'rose', 'plant', 'app
le')
```

Remove Items

In [93]:
```
thistuple = ("apple", "banana", "cherry")
y = list(thistuple)
y.remove("apple")
thistuple = tuple(y)
```

```
print(thistuple)
```

```
('banana', 'cherry')
```

Or you can delete the tuple completely:

In [96]:
```
thistuple = ("apple", "banana", "cherry")
thistuple
```

Out[96]:
```
('apple', 'banana', 'cherry')
```

In [98]:
```
del thistuple
```

In [100…
```
print(thistuple)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[100], line 1
----> 1 print(thistuple)

NameError: name 'thistuple' is not defined
```

Python - Unpack Tuples

In [103…
```
# When we create a tuple, we normally assign values to it. This is called
# Ex: fruits = ("apple", "banana", "cherry")
# But, in Python, we are also allowed to extract the values back into var
```

In [107…
```
# Example
fruits = ("apple", "banana", "cherry")
fruits
```

Out[107…
```
('apple', 'banana', 'cherry')
```

In [109…
```
(green, yellow, red) = fruits
print(fruits)
```

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

In [111…
```
print(green)
print(yellow)
print(red)
```

```
apple
banana
cherry
```

In [113…
```
# te: The number of variables must match the number of values in the tupl
```

```
# you must use an asterisk to collect the remaining values as a list.
```

Using Asterisk*

In [116…
```
# If the number of variables is less than the number of values, you can a
# name and the values will be assigned to the variable as a list:
```

In [118…
```
fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")
(green, yellow, red) = fruits
print(fruits)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[118], line 2
      1 fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")
----> 2 (green, yellow, red) = fruits
      3 print(fruits)

ValueError: too many values to unpack (expected 3)
```

In [120…
```
fruits = ("apple", "banana", "cherry", "strawberry", "raspberry")
fruits
```

Out[120… ('apple', 'banana', 'cherry', 'strawberry', 'raspberry')

In [124…
```
(green, yellow, *red) = fruits
print(green)
print(yellow)
print(red)
```

```
apple
banana
['cherry', 'strawberry', 'raspberry']
```

Python - Loop Tuples

In [127…
```
thistuple = ("apple", "banana", "cherry")
thistuple
```

Out[127… ('apple', 'banana', 'cherry')

In [135…
```
for x in thistuple:
    print (x)
```

```
apple
banana
cherry
```

Loop Through the Index Numbers

In [138…
```
thistuple = ("apple", "banana", "cherry")
for i in range(len(thistuple)):
  print(thistuple[i])
```

```
apple
banana
cherry
```

Using a While Loop

In [145…
```
thistuple = ("apple", "banana", "cherry")
i =0
while i < len(thistuple):
    print(thistuple[i])
    i = i + 1
```

```
apple
banana
cherry
```

Python - Join Tuples

In [148…
```
tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)
tuple3 = tuple1 + tuple2
print(tuple3)
```

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

In [150…
```
tuple1 = ("John", "Peter", "Vicky")
newT = "-".join(tuple1)
print(newT)
```

```
John-Peter-Vicky
```

Multiply Tuples

In [153…
```
tuple1 = ("John", "Peter", "Vicky")
tuple1
```

Out[153… ('John', 'Peter', 'Vicky')

In [155…
```
tuple2 = tuple1*2
```

In [157…
```
print(tuple2)
```

```
('John', 'Peter', 'Vicky', 'John', 'Peter', 'Vicky')
```

Python Tuple count() Method

In [161…
```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
thistuple
```

Out[161… (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

In [163…
```
thistuple.count(5)
```

Out[163… 2

Python Tuple index() Method

In [166…
```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
thistuple
```

Out[166… (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

In [170…
```
indx = thistuple.index(4)
print(indx)
```

```
6
```

In [172…
```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
indx = thistuple.index(10)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[172], line 2
      1 thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)
----> 2 indx = thistuple.index(10)

ValueError: tuple.index(x): x not in tuple
```

In [ ]:

In [ ]: