TUPLE_VHA

December 7, 2023

1 Tuples

A tuple is a sequence of immutable or unchangeable Python objects, like lists. The difference between tuples and lists is that tuples cannot be changed. Additionally, lists use square brackets, whereas tuples use parentheses.

2 We will cover the following topics in this chapter:

Creating a tuple

Accessing values in tuples

Updating tuples

Deleting tuple elements

Basic tuple operations

Indexing, slicing, and matrices

Built-in tuple functions

3 Characterstics

Ordered

Unchangeble

Allows duplicate

4 Creating Tuple

Creating a tuple is as simple as separating values using commas. You may also put these comma-separated values between parentheses.

[1]:
```
# empty
t1 = ()
print(t1)
print(type(t1))
# create a tuple with a single item
t2 = ('hello',)
```

```
print(t2)
print(type(t2))
# homo
t3 = (1,2,3,4)
print(t3)
# hetro
t4 = (1,2.5,True,[1,2,3])
print(t4)
# tuple
t5 = (1,2,3,(4,5))
print(t5)
# using type conversion
t6 = tuple('hello')
print(t6)
```

```
()
('hello',)
<class 'tuple'>
(1, 2, 3, 4)
(1, 2.5, True, [1, 2, 3])
(1, 2, 3, (4, 5))
('h', 'e', 'l', 'l', 'o')
```

[2]:
```
t1=(1)
print(type(t1))
```

```
<class 'int'>
```

[4]:
```
#ordered
(1,2,3)==(3,2,1)
```

[4]:
```
False
```

[5]:
```
#unchangable
t=(1,2,3)
t[0]=4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27756\531563010.py in <module>
      1 #unchangable
      2 t=(1,2,3)
----> 3 t[0]=4

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

[6]:
```
#duplicate
t=(3,1,2,1,3,2)
print(t)
```

```
(3, 1, 2, 1, 3, 2)
```

5 Accessing values in tuples

Use the square brackets to access values in a tuple and the index or indices to obtain a value of that index.

Indexing

Slicing

[7]:
```
#indexing
t3 = (1,2,3,4)
print(t3)
prnt(t3[0])
print(t3[-1])
```

```
(1, 2, 3, 4)
1
4
```

[10]:
```
#slicing
t3 = (1,2,3,4)

print(t3)
print(t3[0:4])
print(t3[-1:-4:-1])
```

```
(1, 2, 3, 4)
(1, 2, 3, 4)
(4, 3, 2)
```

6 Updating tuples (edit)

A tuple cannot be changed once it is created, so tuples are immutable.

However, there is a workaround. We can convert a tuple into a list, change it, and then convert it back into a tuple.

[11]:
```
t1=(1,2,3,4)
t1[0]=5
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27756\1107565111.py in <module>
      1 t1=(1,2,3,4)
```

```
----> 2 t1[0]=5

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

7 adding tuple not possible

8 Deleting tuple elements

Individual elements from a tuple cannot be removed, but there is nothing wrong with constructing another tuple without the unwanted elements.

Use the del statement to explicitly remove a tuple

[13]:
```
t3=(1,2,3,4)
print(t3)
del t3
print(t3)
```

```
(1, 2, 3, 4)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27756\1737609542.py in <module>
      1 print(t3)
      2 del t3
----> 3 print(t3)

NameError: name 't3' is not defined
```

[15]:
```
t3=(1,2,3,4)
print(t3)
del t3[2]
print(t3)
```

```
(1, 2, 3, 4)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27756\3802380517.py in <module>
      1 t3=(1,2,3,4)
      2 print(t3)
----> 3 del t3[2]
      4 print(t3)

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

9 Operations on Tuples

Arithmatic

Membership

loop

compare

[16]:
```
# + and *
t1 = (1,2,3,4)
t2 = (5,6,7,8)

print(t1 + t2)

print(t1*3)
# membership
1 in t1
# iteration
for i in t1:
  print(i)
for i,j in enumerate(t1):
    print(i,j)
```

```
(1, 2, 3, 4, 5, 6, 7, 8)
(1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4)
1
2
3
4
0 1
1 2
2 3
3 4
```

[55]:
```
print((1,2,3)==(1,2,3,4))
print((1,2,"a")==(1,2,"A"))
print((1,2,"a")>(1,2,"A"))
print(("v",2,"a")>(1,2,"A"))
```

```
False
False
True
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27756\2933097873.py in <module>
      2 print((1,2,"a")==(1,2,"A"))
      3 print((1,2,"a")>(1,2,"A"))
```

```
----> 4 print(("v",2,"a")>(1,2,"A"))

TypeError: '>' not supported between instances of 'str' and 'int'
```

10 Tuple in-built function

11 Len

This function is used to determine the length of the tuple.

Syntax:

len(tuple)

[18]:
```
t=(1,2,3,4)
len(t)
```

[18]:
```
4
```

12 Tuple max() method

The max() method returns the elements from the tuple with maximum value.

Syntax:

max(tuple)

Parameters

tuple - This is a tuple from which the max valued element is to be returned.

Return value

This method returns the elements from the tuple with maximum value.

[20]:
```
t=(1,2,3,4)
max(t)
```

[20]:
```
4
```

[21]:
```
t=(1,2,3,4,"a")
max(t)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27756\2127476705.py in <module>
      1 t=(1,2,3,4,"a")
----> 2 max(t)
```

```
TypeError: '>' not supported between instances of 'str' and 'int'
```

13 Tuple min() method

The min() method returns the elements from the tuple with minimum value.

Syntax:

min(tuple) Parameters

tuple - This is a tuple from which min valued element is to be

returned.

Return value

This method returns the elements from the tuple with minimum value.

[24]:
```
t=(1,2,3,4)
min(t)
```

[24]: 1

[25]:
```
t=(1,2,3,4,'a')
min(t)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27756\3396944552.py in <module>
      1 t=(1,2,3,4,'a')
----> 2 min(t)

TypeError: '<' not supported between instances of 'str' and 'int'
```

14 Tuple tuple() method

The tuple() method converts a list of items into tuples.

Syntax:

tuple(seq)

Parameters

seq - This is a tuple to be converted into tuple. Return value

[26]:
```
t=tuple(range(1,10))
print(t)
```

```
(1, 2, 3, 4, 5, 6, 7, 8, 9)
```

[27]:
```
t=tuple([1,2,3,4])
print(t)
```

```
(1, 2, 3, 4)
```

[29]:
```
t=tuple("hello")
print(t)
```

```
('h', 'e', 'l', 'l', 'o')
```

15 Tuple function

sorted

count

index

sum

reversed

enumerate

[30]:
```
t=(1,5,2,8,3)
print(sorted(t))
print(sorted(t,reverse=True))
print(sorted(t,reverse=False))
```

```
[1, 2, 3, 5, 8]
[8, 5, 3, 2, 1]
[1, 2, 3, 5, 8]
```

[31]:
```
t=(1,1,2,3,4,1,5,8)
t.count(1)
```

[31]:
```
3
```

[32]:
```
t=(1,1,2,3,4,1,5,8)
t.index(1)
```

[32]:
```
0
```

[33]:
```
t=(1,1,2,3,4,1,5,8)
t.index(1,3,7)
```

[33]:
```
5
```

[34]:
```
t=(1,1,2,3,4,1,5,8)
sum(t)
```

```
[34]: 25
```

```
[2]: t=(1,1,2,3,4,1,5,8)
     tuple(reversed(t))
```

```
[2]: (8, 5, 1, 4, 3, 2, 1, 1)
```

```
[4]: t=(1,1,2,3,4,1,5,8)
     tuple(enumerate(t))
```

```
[4]: ((0, 1), (1, 1), (2, 2), (3, 3), (4, 4), (5, 1), (6, 5), (7, 8))
```

```
[6]: t=(1,1,2,3,4,1,5,8)
     tuple(enumerate(t,start=10))
```

```
[6]: ((10, 1), (11, 1), (12, 2), (13, 3), (14, 4), (15, 1), (16, 5), (17, 8))
```

16 Difference between Lists and Tuples

Syntax

Mutability

Speed

Memory

Built in functionality

Error prone

Usability

```
[35]: import time

      L = list(range(100000000))
      T = tuple(range(100000000))

      start = time.time()
      for i in L:
        i*5
      print('List time',time.time()-start)

      start = time.time()
      for i in T:
        i*5
      print('Tuple time',time.time()-start)
```

```
List time 4.274397134780884
Tuple time 4.468472003936768
```

[36]:
```
import time

L = list(range(100000000))
start = time.time()
for i in L:
  i*5
print('List time',time.time()-start)
```

```
List time 4.362755060195923
```

[37]:
```
import time

T = tuple(range(100000000))
start = time.time()
for i in T:
  i*5
print('Tuple time',time.time()-start)
```

```
Tuple time 4.776736736297607
```

[38]:
```
import sys

L = list(range(1000))
T = tuple(range(1000))

print('List size',sys.getsizeof(L))
print('Tuple size',sys.getsizeof(T))
```

```
List size 8056
Tuple size 8040
```

[39]:
```
a = [1,2,3]
b = a

a.append(4)
print(a)
print(b)
```

```
[1, 2, 3, 4]
[1, 2, 3, 4]
```

[40]:
```
a = (1,2,3)
b = a

a = a + (4,)
print(a)
print(b)
```

```
(1, 2, 3, 4)
(1, 2, 3)
```

17 Why use tuple?

18 Special Syntax

[41]:
```
# tuple unpacking
a,b,c = (1,2,3)
print(a,b,c)
```

```
1 2 3
```

[42]:
```
a,b = (1,2,3)
print(a,b)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_27756\164063033.py in <module>
----> 1 a,b = (1,2,3)
      2 print(a,b)

ValueError: too many values to unpack (expected 2)
```

[43]:
```
a = 1
b = 2
a,b = b,a

print(a,b)
```

```
2 1
```

[44]:
```
a,b,*others = (1,2,3,4)
print(a,b)
print(others)
```

```
1 2
[3, 4]
```

[45]:
```
# zipping tuples
a = (1,2,3,4)
b = (5,6,7,8)

tuple(zip(a,b))
```

[45]:
```
((1, 5), (2, 6), (3, 7), (4, 8))
```

19 Q1: Join Tuples if similar initial element

While working with Python tuples, we can have a problem in which we need to perform concatenation of records from the similarity of initial element. This problem can have applications in data domains such as Data Science.

For eg.

Input : test_list = [(5, 6), (5, 7), (5, 8), (6, 10), (7, 13)] Output : [(5, 6, 7, 8), (6, 10), (7, 13)]

[46]:
```
test_list = [(5, 6), (5, 7), (5, 8), (6, 10), (7, 13)]

unique = []

for i in test_list:
  unique.append(i[0])
unique = set(unique)

result = []
for i in unique:
  result.append([i])
  for j in test_list:
    if j[0] == i:
      result[-1].append(j[1])

print(list(map(tuple,result)))
```

```
[(5, 6, 7, 8), (6, 10), (7, 13)]
```

20 Q2: Multiply Adjacent elements (both side) and take sum of right and lest side multiplication result.

For eg.

The original tuple : (1, 5, 7, 8, 10)

Resultant tuple after multiplication :

(15, 15+57, 75 + 78, 87 + 810, 108) -> (5, 40, 91, 136, 80) output-(5, 40, 91, 136, 80)

[47]:
```
# write your code here
t = (1, 5, 7, 8, 10)

L = []

L.append(t[0]*t[1])

for i in range(1,len(t)-1):
  L.append(t[i]*t[i-1] + t[i]*t[i+1])

L.append(t[-1]*t[-2])

print(tuple(L))
```

```
(5, 40, 91, 136, 80)
```

21 Q3: Check is tuples are same or not?

Two tuples would be same if both tuples have same element at same index

t1 = (1,2,3,0)

t2 = (0,1,2,3)

t1 and t2 are not same

[48]:
```
# write your code here
t1 = (1,2,3,0)
t2 = (1,2,3,0)

flag = True
for i,j in zip(t1,t2):
  if i == j:
    continue
  else:
    flag = False
    break
if flag:
  print('same')
else:
  print('not same')
```

```
same
```

22 Q4: Count no of tuples, list and set from a list

list1 = [{‘hi’, ‘bye’},{‘Geeks’, ‘forGeeks’},(‘a’, ‘b’),[‘hi’, ‘bye’],[‘a’, ‘b’]]

Output:

List-2

Set-2

Tuples-1

[49]:
```
# write your code here
L = [{'hi', 'bye'},{'Geeks', 'forGeeks'},('a', 'b'),['hi', 'bye'],['a', 'b']]
output = [0,0,0]

for i in L:
  if type(i) == list:
    output[0] = output[0] + 1
  elif type(i) == set:
    output[1] = output[1] + 1
  elif type(i) == tuple:
    output[2] = output[2] + 1
  else:
    pass

print('Lists-{}\nSets-{}\nTuples-{}'.format(output[0],output[1],output[2]))
```

```
Lists-2
Sets-2
Tuples-1
```

23 Q5: Shortlist Students for a Job role

Ask user to input students record and store in tuples for each record. Then Ask user to input three things he wants in the candidate- Primary Skill, Higher Education, Year of Graduation.

Show every students record in form of tuples if matches all required criteria.

It is assumed that there will be only one primry skill.

If no such candidate found, print No such candidate

Input:

Enter No of records- 2

Enter Details of student-1

Enter Student name- Manohar

Enter Higher Education- B.Tech

Enter Primary Skill- Python

Enter Year of Graduation- 2022

Enter Details of student-2

Enter Student name- Ponian

Enter Higher Education- B.Sc.

Enter Primary Skill- C++

Enter Year of Graduation- 2020

Enter Job Role Requirement

Enter Skill- Python

Enter Higher Education- B.Tech

Enter Year of Graduation- 2022

Output

('Manohar', 'B.tech', 'Python', '2022')

[50]:
```python
# write your code here
students = []

num = int(input('enter the number of applicants'))

for i in range(num):
  print('Enter details of',i+1,'applicant:')
  name = input('enter name')
  h_ed = input('enter higher education')
  p_skill = input('enter primary skill')
  yog = input('enter year of graduation')

  students.append((name,h_ed,p_skill,yog))

required_skill = input('enter required skill')
required_hed = input('enter required higher education')
required_yog = input('enter required year of graduation')

flag = False
for i in students:
  if i[1] == required_hed and i[2] == required_skill and i[3] == required_yog:
    print(i)
    flag = True

if flag == False:
  print('No such candidates')
```

```
enter the number of applicants2
Enter details of 1 applicant:
enter namekavit
enter higher educationPh.D
enter primary skillpython
enter year of graduation2016
Enter details of 2 applicant:
enter namemanish
enter higher educationPh.D
enter primary skillc++
enter year of graduation2005
enter required skillpython
enter required higher educationB.Tech
enter required year of graduation2022
No such candidates
```

[ ]:
```

```