ED5340 - Data Science: Theory and Practise

L5 - Tuple container

Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

Tuple representation

- Tuples is a collection of heterogeneous data types enclosed in C brackets
  - empty_tup = tup( ) #empty tuple
  - tup1 = (10,) # Note that you need a comma
  - tup2 = (‘Ram’, 20, 33.333)

Tuple basics and accessing

- Elements in the tuple can be repeated.
- It is also a sequential collection - can be indexed and sliced
- Entire tuple or each element can be printed.
- Tuple is an `iterable' i.e. you can iterate over its elements.

Demo using L6_tuple_ex_access.py

CW: Define a tuple of different datatypes and print them using iterator.

HW:In a tuple, find the number of objects of each type.

Tuple operations

- Tuples are immutable (like strings).
- Tuples can be concatenated
- searching (containment) and sorting
- deletion - using index or range of indices
- conversion / other functions - str to tuple, len, max, min, sum
- index and count (member functions)
- tuple comparison

Demo using L6_tuple_operations.py

HW: Find out the functions that are not in tuples but available in lists and why?

Some points on Tuples

- Tuple is an iterable.
- Tuple is like a structure in C.
- No tuple comprehension (why?)
- You can use list comprehension and then use ‘tuple’ function

Tuple Varieties

Similar to List varieties

- Tuple of Tuples
- Tuple embedding
- Tuple unpacking (using * operator)

Grouping Tuples

Taking one each from each tuple

names = ('Ram', 'Raja', 'Geetha', 'Ramya')

gender = ('male', 'male', 'female', ‘female')

You want to group the tuples to the following:

('Ram', 'male'), ('Raja', 'male'), ('Geetha', 'female'), ('Ramya', 'female')

(names[0], gender[0]), (names[1], gender[1]), and so on

Grouping Tuples

Taking one each from each tuple

fruits = ('apples', 'oranges', 'grapes', 'guava')

num_kg = (2, 5, 3, 6)

cost_kg = (150, 80, 200, 170)

Group the tuples

('apples', 2, 150), …….

(fruits[0], num_kg[0], cost_kg[0]), ……..

zip( ) function

grouping can be achieved by a zip( ) function

- Takes one or more iterables and groups them together
- returns an iterator of tuples
- zip( ) function - a very important one

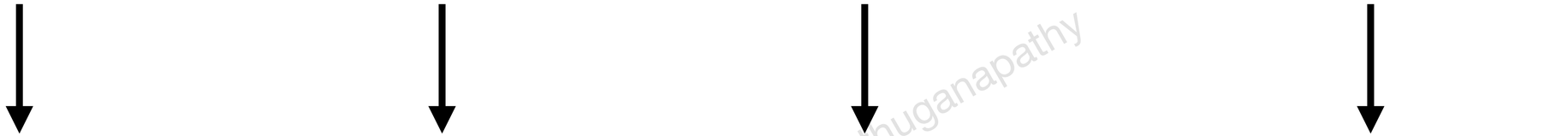
zip( ) function
grouping tuples

```
names = ('Ram', 'Raja', 'Geetha', 'Ramya')

gender = ('male', 'male', 'female', ‘female’)

#Sevaral ways in which zip( ) can be used

ite = zip(names, gender) #zip returns an iterator of tuples

print(*ite)  #Remember that ite is like a pointer and hence * is needed to unpack

#You can also do the following

for ite in zip(names, gender):

    print(*ite)
```

Some nuances on iterator of zip

we will see more on iterators later!

('Ram', 'male'), ('Raja', 'male'), ('Geetha', 'female'), ('Ramya', 'female')



ite ite = ite->next ite = ite->next ite = ite->next

For printing

*ite *ite *ite *ite

ite[0], ite[1] ite[0], ite[1] ite[0], ite[1] ite[0], ite[1]

Note: In Python, ite -> next is actually notated as ite.__next__( )

CW: Take three lists, one each for few names, their ages and salaries and make a tuple out of the lists.

HW: WAP to print the transpose of a 3X5 matrix.

Matrix problems using zip( )

mat = [[1,2,3], [4,5,6]]

zip(mat[0], mat[1]) will give three tuples
(1,4) (2,5) (3,6) (needs iterator to fetch them)


```
for ite in zip(mat[0], mat[1]):
        print(ite)  #prints each tuple
```

```
for ite in zip(*mat):
      print(ite)  #prints each tuple
```

zip(*mat) is same as
zip(mat[0], mat[1]

Matrix problems using zip( )

Inverse

```
mat = [[1,2,3], [4,5,6]]
ite = zip(*mat) #gives (1, 4), (2,5), (3, 6)
lst = list(ite)
print(lst)
```