**Namedtuple in Python**

Python supports a type of container like [dictionaries](http://quiz.geeksforgeeks.org/python-set-4-dictionary-keywords-python/) called “**namedtuple()**” present in module, “[**collections**](https://www.geeksforgeeks.org/python-collections-module/)“. Like dictionaries they contain keys that are hashed to a particular value. But on contrary, it supports both access from key value and iteration, the functionality that dictionaries lack.

**Example:**

# Python code to demonstrate namedtuple()

from collections import namedtuple

# Declaring namedtuple()

Student = namedtuple('Student',['name','age','DOB'])

# Adding values

S = Student('Dharma','20','18032001')

# Access using index

print ("The Student age using index is : ",end ="")

print (S[1])

# Access using name

print ("The Student name using keyname is : ",end ="")

print (S.name)

## Conversion Operations

* **\_make() :-** This function is used to return a **namedtuple() from the iterable** passed as argument.
* **\_asdict() :-** This function returns **the** [**OrderedDict()**](https://www.geeksforgeeks.org/ordereddict-in-python/) as constructed from the mapped values of namedtuple().
* **using “\*\*” (double star) operator** :- This function is used to **convert a dictionary into the namedtuple().**

# Python code to demonstrate namedtuple() and

# \_make(), \_asdict() and "\*\*" operator

# Declaring namedtuple()

Student = collections.namedtuple('Student',['name','age','DOB'])

# Adding values

S = Student('Dharma','20','18032001')

# initializing iterable

li = ['Manjeet', '19', '411997' ]

# initializing dict

di = { 'name' : "Nikhil", 'age' : 19 , 'DOB' : '1391997' }

# using \_make() to return namedtuple()

print ("The namedtuple instance using iterable is  : ")

print (Student.\_make(li))

# using \_asdict() to return an OrderedDict()

print ("The OrderedDict instance using namedtuple is  : ")

print (S.\_asdict())

# using \*\* operator to return namedtuple from dictionary

print ("The namedtuple instance from dict is  : ")

print (Student(\*\*di))