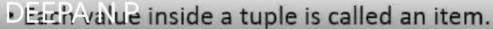


Tuples (): Creating Tuples

- In mathematics, a tuple is a finite ordered list (sequence) of elements.
 A tuple is defined as a data structure that comprises an ordered, finite
 sequence of immutable, heterogeneous elements that are of fixed
 sizes.
- A tuple is a finite ordered list of values of possibly different types which is used to bundle related values together without having to create a specific type to hold them.
- Tuples are immutable
- Once a tuple is created, you cannot change its values. A tuple is defined by putting a comma-separated list of values inside parentheses ().





Basic Tuple Operations

```
>>> tuple_items = (1, 9, 8, 8)
>>> 1 in tuple_items

True
>>> 25 in tuple_items

False
>>> tuple_1 = (9, 8, 7)
>>> tuple_2 = (9, 1, 1)
>>> tuple_1 > tuple_2

True
>>> tuple_1 != tuple_2

True
>>> tuple_1 != tuple_2
```

Built-In Functions Used on Tuples

Built-In Functions Used on Tuples

>>> sorted years

Built-In Functions	Description	
len()	The len() function returns the numbers of items in a tuple.	
sum()	The sum() function returns the sum of numbers in the tuple	
sorted()	The serted() function returns a sorted copy of the tuple as a list while leaving the original tuple unfouched.	

```
For example, 
>>> years = (1987, 1985, 1981, 1996) 
>>> len(years) 
>>> sorted_years = sorted(years)
```

If an item within a tuple is mutable, then you can change it. Consider the presence of a list as an item in a tuple, then any changes to the list get reflected on the overall items inthe tuple. For example,

- 1. >>> german_cars = ["porsche", "audi", "bmw"]
- 2. >>> european_cars = ("ferrari", "volvo", "renault", german_cars)
- 3. >>> european_cars
- ('ferrari', 'volvo', 'renault', ['porsche', 'audi', 'bmw'])
- 4. >>> european_cars[3].append("mercedes")
- 5. >>> german cars
- ['porsche', 'audi', 'bmw', 'mercedes']
- 6. >>> european_cars
- ('ferrari', 'volvo', 'renault', ['porsche', 'audi', 'bmw', 'mercedes'])

DEEPA N P

Tuple Methods >>> dir(tuple)

Various Tuple Methods			
Tuple Methods	Syntax	Description	
count()	tuple_name.count(item)	The count() method counts the number of times the item has occurred in the tuple and returns it.	
index()	tuple_name.index(item)	The index() method searches for the given item from the start of the tuple and returns its index. If the value appears more than once, you will get the index of the first one. If the item is not present in the tuple, then ValueError is thrown by this method.	

8

```
channels = ("ngc", "discovery", "animal_planet", "history", "ngc")
```

print(channels.count("ngc"))

print(channels.index("history"))

Tuple Packing and Unpacking

The statement t = 12345, 54321, 'hello!' is an example of tuple packing.

```
t = 12005, Ca52), "(wild)"
print(t)
(12045, 5432), "helin")
```

The reverse operation of tuple packing is also possible. This operation is called tuple unpacking and works for any sequence on the right-hand side.

```
x, y, z = mond(x)
mond(y)
mond(y)
print(z)
12161
54721
hello!
```

Using zip() Function

- The zip() function makes a sequence that aggregates elements from each of the iterables (can be zero or more).
- The syntax for zip() function is,

zip(*iterables)

 An <u>iterable</u> can be a list, string, or dictionary. It returns a sequence of tuples, where the <u>i-th</u> tuple contains the <u>i-th</u> element from each of the iterables.

[(1, 4), (2, 5), (3, 6)]