Tuples - Operators and Methods

This notebook covers the operators and methods applicable to tuples.

1. The concatenation (`+`) operator
2. The multiplication (`*`) operator

Tuples have only two methods(). As usual, they are accessed as follows: `tuplename.methodname()`

1. The `count()` method : counts the number of times an element occurs in a list
2. The `index()` method : returns the index of the first occurence of the argument

We also discuss tuple assignment.

We can concatenate two tuples together using the concatenation (`+`) operator.

In [3]:

```
a = (1, 2, 3)
b = (4,)
c = a + b
print(c)
```

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

We can use the multiplication operator (`*`) to repeat a tuple multiple times.

In [4]:

```
aTuple = ('xyz', 'zara', 'abc', 'xyz')
newTuple = aTuple*2
newTuple2 = 2*aTuple # The int value can come before or after the tuple
print(aTuple)
print(newTuple)
print(newTuple2)
```

```
('xyz', 'zara', 'abc', 'xyz')
('xyz', 'zara', 'abc', 'xyz', 'xyz', 'zara', 'abc', 'xyz')
('xyz', 'zara', 'abc', 'xyz', 'xyz', 'zara', 'abc', 'xyz')
```

The `count()` method counts the number of times an element occurs in a tuple.

In [5]:

```
aTuple = (123, 'xyz', 'zara', 'abc', 123)
print("Count for 123 : ", aTuple.count(123))
print("Count for zara : ", aTuple.count('zara'))
```

```
Count for 123 :  2
Count for zara :  1
```

The `index()` method returns the index of the first occurence of the argument.

In [6]:

```
aTuple = (123, 'xyz', 'zara', 'abc', 123)
print("Index for xyz : ", aTuple.index( 'xyz'))
print("Index for zara : ", aTuple.index( 'zarak')) # Will return an error since 'zarak' is not in
the tuple.
```

```
Index for xyz :  1
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-6-5dc3547cff2f> in <module>
      1 aTuple = (123, 'xyz', 'zara', 'abc', 123)
      2 print("Index for xyz : ", aTuple.index( 'xyz'))
----> 3 print("Index for zara : ", aTuple.index( 'zarak')) # Will return an error since 'zarak' is not in the tuple.

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

You can use tuple assignment to assign values to multiple variables in a single Python statement. Note that the number of variables must exactly equal the number of values. Note also that we can omit the parentheses in tuple assignment.

In [7]:

```
aTuple = (123, 'xyz', 'zara', 'abc', 123)
a,b,c,d,e = aTuple
print(a,b,c,d,e)
```

```
123 xyz zara abc 123
```

You can use slicing if you wish to use tuple assignment to assign values to a fewer number of variables.

In [8]:

```
aTuple = (123, 'xyz', 'zara', 'abc', 123)
a,b,c = aTuple[:3]
print(a,b,c)
```

```
123 xyz zara
```