Tuple

Lists and tuples have many similarities. Some of them have been enlisted below:

- They are both sequence data types that store a collection of items
- They can store items of any data type
- And any item is accessible via its index.

How lists and tuples differ from each other.

Syntax Difference

A list is created using square brackets [] whereas the tuple is created using parenthesis ().

```
tuple_names = ('Nicholas', 'Michelle', 'Alex')

list_names = ['Nicholas', 'Michelle', 'Alex']

print(tuple_names)

print(list_names)
```

Python has the **type()** object that helps us know the type of an object that has been created. We can use it as follows:

```
print(type(tuple_names))

print(type(list_names))
```

Mutable vs. Immutable

Lists are mutable while tuples are immutable. We can change/modify the values of a list but we cannot change/modify the values of a tuple.

```
names = ["Nicholas", "Michelle", "Alex"]

names[0] = "Samuel"

names = ("Nicholas", "Michelle", "Alex")

names[0] = "Samuel"
```

A tuple object cannot be changed after it has been created. So Tuple can be used as a key for dictionary…

Size Difference

Tuple will have a smaller memory compared to the list.

```
tuple_names = ('Nicholas', 'Michelle', 'Alex')

list_names = ['Nicholas', 'Michelle', 'Alex']

print(tuple_names.__sizeof__())

print(list_names.__sizeof__())
```

Variable Length vs. Fixed Length

Tuples have a fixed length while lists have a variable length. This means we can change the size of a created list but we cannot change the size of an existing tuple.

Access Tuple Items

```
thistuple = ("apple", "banana", "cherry")

print(thistuple[1])
```

Negative Indexing

```
thistuple = ("apple", "banana", "cherry")

print(thistuple[-1])
```

Range of Indexes

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")

print(thistuple[2:5])
```

Range of Negative Indexes

```
thistuple = ("apple", "banana", "cherry", "orange", "kiwi", "melon", "mango")
print(thistuple[-4:-1])
```

Loop Through a Tuple

```
thistuple = ("apple", "banana", "cherry")
for x in thistuple:
  print(x)
```

Check if Item Exists

```
thistuple = ("apple", "banana", "cherry")
if "apple" in thistuple:
  print("Yes, 'apple' is in the fruits tuple")
```

Tuple Length

```
thistuple = ("apple", "banana", "cherry")
print(len(thistuple))
```

Create Tuple With One Item

```
thistuple = ("apple",)
print(type(thistuple))

#NOT a tuple
thistuple = ("apple")
print(type(thistuple))
```

The del keyword can delete the tuple completely:

```
thistuple = ("apple", "banana", "cherry")
del thistuple
print(thistuple)
```

Join Two Tuples

```
tuple1 = ("a", "b" , "c")
tuple2 = (1, 2, 3)
tuple3 = tuple1 + tuple2
print(tuple3)
```

Tuple count() Method

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

x = thistuple.count(5)

print(x)
```

Tuple index() Method

```
thistuple = (1, 3, 7, 8, 7, 5, 4, 6, 8, 5)

x = thistuple.index(8)

print(x)
```