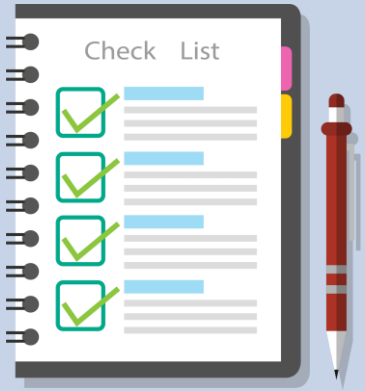
Python – Basics

Digital Lync

INNOVATION - EDUCATION - INCUBATION

Shah Ayub Quadri
aquadri@digital-lync.com

Index



Python Basics

Tuple

- Introduction
- Accessing Tuples
- Operations
- Working & Tuple Comprehension
- Functions and Methods

Tuple

- A tuple is a sequence of **immutable** Python objects. Tuples are sequences, just like lists.
- The differences between tuples and lists are, the tuples cannot be changed unlike lists and **tuples use parentheses**, whereas **lists use square brackets**.

Examples of Tuple :

```
tup1 = ('A', 'B', 1, 2)
tup2 = (1, 2, 3, 4, 5 )
tup3 = "a", "b", "c", "d";
print(tup1)
print(tup2)
print(tup3)
```

```
C:\Users\ravikiran\Desktop\python>a.py
('A', 'B', 1, 2)
(1, 2, 3, 4, 5)
('a', 'b', 'c', 'd')
```

Accessing Values in Tuples:

- To access values in tuple, use the square brackets for slicing along with the index or indices to obtain value available at that index.

For example:

```
tup = (1, 2, 3, 4, 5,6,7,8,9 )

print ("tup1[0]: ", tup[0])
print ("tup2[1:5]: ", tup[1:5])
```

```
C:\Users\ravikiran\Desktop\python>a.py
tup1[0]:  1
tup2[1:5]:  (2, 3, 4, 5)
```

Tuple

Updating Tuple:

- Tuples are immutable which means you cannot update or change the values of tuple elements.
- One can take portions of existing tuples to create new tuple.

- **For example:**

```
tup = (12, 34);
tup1 = ('abc', 'xyz');
#tup1[0] = 100;    Following action is not valid for tuples
tup2 = tup + tup1;
print (tup2)
```

```
C:\Users\ravikiran\Desktop\python>a.py
(12, 34, 'abc', 'xyz')
```

Tuple

Delete Tuple Elements :

- Removing individual tuple elements is not possible. There is, of course, nothing wrong with putting together another tuple with the undesired elements discarded.
- To explicitly remove an entire tuple, just use the **del** statement.

For example:

```
t=(1,2,3,4,5,6,7,8,9)

print (t)
del (t)
print ("After deleting tup : ")
print (t)
```

```
C:\Users\ravikiran\Desktop\python>a.py
(1, 2, 3, 4, 5, 6, 7, 8, 9)
After deleting tup :
Traceback (most recent call last):
  File "C:\Users\ravikiran\Desktop\python\a.py", line 24, in <module>
    print (t)
NameError: name 't' is not defined
```

Basic Tuples Operations :

Python Expression	Results	Description
len((1, 2, 3))	3	Length
(1, 2, 3) + (4, 5, 6)	(1, 2, 3, 4, 5, 6)	Concatenation
('Hi!',) * 4	('Hi!', 'Hi!', 'Hi!', 'Hi!')	Repetition
3 in (1, 2, 3)	True	Membership
for x in (1, 2, 3): print x,	1 2 3	Iteration

Built-in Tuple Functions:

.

- **cmp(tuple1, tuple2):** The method **cmp()** compares elements of two tuple.
- **len(tuple):** The method **len()** returns the number of elements in the tuple.
- **Max (tuple) :** The method **max()** returns the elements from the tuple with maximum value.
- **min(tuple) :** The method **min()** returns the elements from the tuple with minimum value.

Tuple Slices:
Slices work on tuples just as with strings, and can also be used to change sub-parts of the tuple.

```
t=(1,2,3,4,5,6,7)

print(t[1])
print(t[:])
print(t[1:5])
print(t[:-1])
```

```
C:\Users\ravikiran\Desktop\python>a.py
2
(1, 2, 3, 4, 5, 6, 7)
(2, 3, 4, 5)
(1, 2, 3, 4, 5, 6)
```

Tuple

Convert a list to a tuple:

- The method **tuple()** converts a list of items into tuples

```
list=[1,2,3,4,5,6,7]
print("list",list)
tuple=tuple(list)
print("tuple",tuple)
```

```
C:\Users\ravikiran\Desktop\python>a.py
list [1, 2, 3, 4, 5, 6, 7]
tuple (1, 2, 3, 4, 5, 6, 7)
```

Assignment - 7

1. Write a Python program to create a tuple.

2. Write a Python program to create a tuple with different data types..

3. Write a Python program to convert a list to a tuple.

4. Write a Python program to slice a tuple.

5. Write a Python program to replace last value of tuple in a list.
   Sample list: [(10, 20, 40), (40, 50, 60), (70, 80, 90)]
   Expected Output: [(10, 20, 100), (40, 50, 100), (70, 80, 100)]