PYTHON

From Simple to Complex
With
Examples

AYESHA NOREEN

Bachelor's in Software Engineering,

Master's in Computer Science

from COMSATS University, Islamabad

NOTE!!!

In these notes Screenshots of practice examples and coding are added. The code files are also available in code folder that contain .ipynb files that are created on Jupyter notebook.

Chapter11
Tuples in Python

Tuples are immutable, ordered and allow duplication. Tuples are same as list but list uses square bracket while tuple uses round bracket. Tuples are faster and having greater performance than list. we use tuple when we don't need any change in our data. e.g

Days=('monday','tuesday','wednesday','thursday','friday','saturday','sunday')

We can not append, insert, remove, del or extend in tuples mean no addition or deletion of data. Tuples are immutable while list is mutable. Only count, len and index function can be apply and we can also do slicing in tuples.

- **Tuple methods**

```
numbers=(1,2,3,4,3,3,4)
print(numbers.count(3))   #counts number of 3 is tuple which is 3
print(numbers.index(4))   #prints index of 4 which is 3
print(len(numbers))       #prints length of tuple which is 7
print(numbers[2:5])       #print tuple from 2nd index to 5th-1=4th
```

```
3
3
7
(3, 4, 3)
```

- **Tuple with single element**

If we have only one element in tuple than place comma after that element otherwise it consider as int or string etc. according to data type of element. e.g.

```
number=(1)
print(type(number))     #prints type is int
number=(1,)
print(type(number))     #gives type is tuple
name=('ayesha')
print(type(name))     #gives type is string
name=('ayesha',)
print(type(number))     #gives type is tuple
```

```
<class 'int'>
<class 'tuple'>
<class 'str'>
<class 'tuple'>
```

- **Tuple without parenthesis**

If we write something without square bracket or parenthesis than it is consider as tuple like

```
mixed=1,2,3,4.0,'ayesha'
print(type(mixed))  #prints tuple
```

- **Tuple Unpacking**

Tuple unpacking mean we can assign each element of tuple to some variable than print it separately. This is called tuple unpacking.

- **Tuple unpacking**

```
#tuple unpacking
mixed=(1,2,3,4.0,'ayesha')
num1,num2,num3,num4,string=(mixed)
print(num1)     #prints 1
print(num2)     #prints 2
print(num3)     #prints 3
print(num4)     #prints 4.0
print(string)     #prints ayesha
```

```
1
2
3
4.0
ayesha
```

- **List inside tuple**

```
#list inside tuple
mixed=(1,2,3,4,5,6.0,'ayesha',['ayesha','noreen','sana'])
mixed[7].pop()
print(mixed)  #it prints (1,2,3,4,5,6.0,'ayesha',['ayesha','noreen'])
mixed[7].append('sana rehman')
print(mixed)  #it prints (1,2,3,4,5,6.0,'ayesha',['ayesha','noreen','sana rehman'])
numbers=(1,2,3,4,5)
print(max(numbers))   #prints 5
print(min(numbers))   #prints 1
print(sum(numbers))   #prints 15
```

```
(1, 2, 3, 4, 5, 6.0, 'ayesha', ['ayesha', 'noreen'])
(1, 2, 3, 4, 5, 6.0, 'ayesha', ['ayesha', 'noreen', 'sana rehman'])
5
1
15
```

- **Function returning two parameters is actually returning a tuple**

```
def fun(num1,num2):
    add=num1+num2
    mul=num1*num2
    return add,mul
#print(fun(5,5))    #it prints a tuple(10,25)
addition,multiply=(fun(5,2))
print(type(fun(5,2)))
print(addition)    #prints 7
print(multiply)    #prints 10
```

```
<class 'tuple'>
7
10
```