List

- Inserting values in the list:

```
my_list.insert(3,' Herat')
```

```
my_list
```

```
['Ahmad', 'Saeed', 'Samim', ' Herat', 'Kabul', 98.87, 'Sadiq']
```

- Sorting the values
- (cannot sort between different types)

```
city_names=['Kabul','Herat','Balkh','Kandahar']
```

```
city_names
```

```
['Kabul', 'Herat', 'Balkh', 'Kandahar']
```

```
city_names.sort()
```

```
city_names
```

```
['Balkh', 'Herat', 'Kabul', 'Kandahar']
```

Dictionaries

- Dictionaries are different than sequences
- They are called Mappings
- They store objects by *key* instead of by relative position normally used to refer to the attributes and object of values stored in databases:
- The Key and Value is separated using “:”
- {‘Key’ : ’value’}

```
price_list={'Tomato': 30, 'Apple': 250, 'Milk': 50}
```

```
price_list
```

```
{'Tomato': 30, 'Apple': 250, 'Milk': 50}
```

- Here, the Tomato, Apple and Milk are keys.

Contd..

- Dictionaries cannot be sorted,
- You don’t have to know about index location

```
price_list
```

```
{'Tomato': 30, 'Apple': 250, 'Milk': 50}
```

```
price_list['Milk']
```

```
50
```

- We can add more keys and values to existent Dictionary:

```
price_list['Orange']=400
price_list
```

```
{'Tomato': 30, 'Apple': 250, 'Milk': 50, 'Orange': 400}
```

- We can find the collection of Keys, and Values
- price_list.Keys()…….. Price_list.values()

Tuples

- Tuples are *sequences*, like lists, but they are *immutable*
- They’re used to represent fixed collections of items

```
my_tuple=(5,2,3,43,232,'Sabawoon')
```

```
my_tuple
```

```
(5, 2, 3, 43, 232, 'Sabawoon')
```

- The other rules of a list is implemented on tuple
- Two specific methods:
  - my_tuple.count(value)
  - My_tuple.index(value)

Sets

- Unordered collections of unique elements

```
my_set=set()

my_set.add(1)
my_set
my_set.add("yes")
my_set
```

```
{1, 'yes'}
```

- We can add repeated values on lists but not on sets

```
my_lists=[1,2,2,2,1,1,2]
my_lists
```

```
[1, 2, 2, 2, 1, 1, 2]
```

```
my_set=set(my_lists)
my_set
```

```
{1, 2}
```