Date: Page:

CHAPTER-6 : DICTIONARY & SETS

- Dictionary is a collection of Key-Value pairs.

→ Syntax:

```
a = { "Key" : "Value",
      "Shayan": "Code",
      "marks": "100",
      "list" : [1,2,9] }
```

(Keys: "Key", "Shayan", "marks", "list"; Values: "Value", "Code", "100", [1,2,9])

a	"Key"	=> Prints "Value"
a	"list"	=> Prints [1,2,9,]

→ Properties of a Python Dictionaries

1. It is Unordered.
2. It is mutable
3. It is indexed.
4. Cannot Contain duplicate keys

→ Dictionary Methods

Consider the following dictionary

```
a = { "name" : "Shayan",
      "from" : "Nawada",
      "marks" : [92,98,96] }
```

1. a.items() : returns a list of (key, value) tuples
2. a.keys() : returns a list containing dictionary's keys
3. a.update({"friend":"Sam"}) : updates the dictionary with supplied key-value pairs

4) a.get("name"): returns the value of the specified keys (and value is released e.g. "Shreyan" is returned here)

→ **Sets in Python**

Set is a collection of non-repetitive elements.

```
S = Set()          => No repetition allowed!
S.add(1)
S.add(2)           => or set = {1, 2}
```

→ **Properties of Sets**

1) Sets are unordered => Element's Order doesn't matter
2) Sets are unindexed => Cannot access elements by index
3) There is no way to change items in sets.
4) Sets cannot contain duplicate values

→ Operation on Sets

Consider the following set:

$S = \{1, 8, 2, 3\}$

1) len(S) : Returns 4, the length of the set

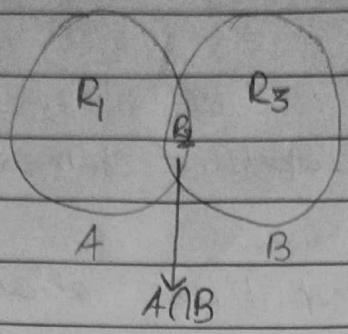
2) S.remove(8): Updates the set S and removes 8 from it

3) S.pop(): Removes an arbitrary element from the set S and returns the element removed

4) S.clear(): empties the set S.

5) S.union({8,11}): Returns a new set with all items from both sets: => {1, 8, 2, 3, 11}

66 5. intersection ({8,11}): Returns a set which contains only items on both sets => {8}



$R_2 \Rightarrow A \cap B$

$R_1 + R_2 + R_3 \Rightarrow A \cup B$

$R_1 + R_3 \Rightarrow A \Delta B$

$R_1 \Rightarrow A - B$

$R_3 \Rightarrow B - A$