Chapter_5 dictionary & sets

Dictionary is a collection of key-value of pairs

Syntax

```
A = { "key"   :   "value"
    "apple"   :   "an apple a day keeps the doctor away"
     "Marks"  :   "50"
     "List"   :   [1, 2, 3]
}
```

A ["key"] ==> prints "value"

A [" list"] ==> prints [1, 2, 3]

Properties of python dictionaries

1. It is unordered
2. It is mutable
3. It is indexed
4. Con not contain duplicate keys

Dictionary methods

Consider the following dictionary

```
a = { "name"      :      "hacksbyte"
      "from"      :     "mars"
      "Number"    :     [1, 2, 3, 4, 5, 6, 7, 8, 9, 0]
}
```

1. a.items() : returns a list of (key value) tuple
2. a.keys() : returns a list containing dictionary keys
3. a.update ({ "friend " : "abc" }) : update the dictionary with supplied key-value pairs
4. a.get ("name") : returns the value is returns
   ("hacksbyte " is returns here)

More methods are available on docs:python.org

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}

If you are a programming beginner without much knowledge of mathematical operations on sets you can simply look at sets in python as data type containing unique values

Properties of sets

1. Sets are unordered ==> element does not 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

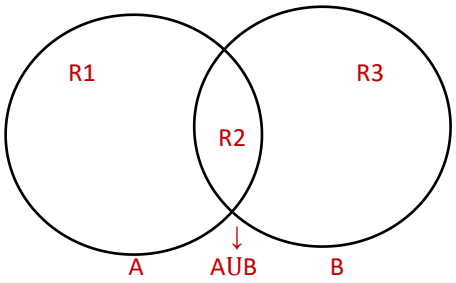
Operations on sets

Consider the following sets

```
S = {1, 2, 4, 8,}
```

1. Len(s) : returns 4 the length of the set
2. S.remove (8) : update the set S and remove 8 from S
3. S.pop() : remove on arbitrary element from the set and returns the element removed

4. S.clear() : empties the set S
5. S.union ({8, 11}) : returns with all items from both sets ==> {1, 2, 4, 8, 11}
6. S.intersection ({8, 11}) : returns a set which contains only items in both sets ==> {8}



R2	=>	A ∩ B
R1+R2+R3	=>	A∪B
R1+R3	=>	A△B
R1	=>	A-B
R3	=>	B-A