SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 OF THE UGC ACT, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

# III B.Tech.

## Computer Science & Engineering

### CSE304: PYTHON PROGRAMMING WITH WEB FRAMEWORKS

### UNIT – II: Dictionaries and Sets

By
Mrs. S. KAMAKSHI, AP-III / CSE
School of Computing

# Dictionary

- **Unordered collections of arbitrary objects as key-value pairs**
- **Accessed by key, not offset position**
  - Also called as associative arrays or hashes
  - Associates a set of values with keys, so an item can be fetched from a dictionary using the key
- **Variable-length, heterogeneous, and arbitrarily nestable**
- **Immutable key but mutable mapping**

# Dictionary

- Embedded within {} as key:value pairs delimited by comma
  - dict_name = {key1:value1, key2:value2, …}
- Eg.

countries = { “TN”:”Tamil Nadu”,
“TE”:”Telungana”,
“AP”: “Andra Pradesh”,
“KE”: “Kerala”,
“KA”: “Karnataka”}

movie = {1960:”Bala Nagamma”, 1961:”Jagathala Pradhapan”, 1963:”Lava Kusa”, 1964:”Karnan”, 1967: “Sri Krishnavatharam”}

# Dictionary Functions


| Method | Description |
|---|---|
| dict_name[key] | Returns the value of the specified key. If key does not exist throws KeyError exception |
| get(key, default_value) | Returns the value of the specified key |
| pop(key, default_value) | Removes the element with the specified key |
| popitem() | Removes the last inserted key-value pair |
| keys() | Returns a view object that contains the dictionary's keys, reflects changes done in dictionary |
| values() | Returns a view object that contains all the values in the dictionary |
| items() | Returns a view object that contains a tuple for each key value pair |
| clear() | Removes all the elements from the dictionary |
| copy() | Returns a copy of the dictionary |
| fromkeys(iterable, [value]) | Returns a dictionary with the specified keys and value |
| setdefault(key, default_value) | Returns the value of the specified key. If the key does not exist: insert the key, with the specified value |
| update(iterable) | Updates the dictionary with the specified key-value pairs |
| del dict_name[key] | Removes the element with the specified key. If key does not exist throws KeyError exception |

# Dictionary

- Retrieving value from dictionary
  - dict_name[key]
  - dict_name.get(key[, value])
  - dict_name.set_default(key[,  value])
- Checking for key in dictionary
  - key in dict_name
- Converting between list and dictionary
  - list(view_object)
  - dict(2D-list or 2D-tuple)

# Set

- Unordered collection of unique and immutable objects that supports operations corresponding to mathematical set theory
- An item appears exactly once no matter how many times it is added to the set
- sets are
  - Iterable
  - Can grow and shrink on demand
  - May contain a heterogeneous types of objects

- To make a set object, pass in a sequence or other iterable object to the built-in set function:
- Eg.
  - x = set('abcde')
  - y = set('bdxyz')
  - x

    {'a', 'c', 'b', 'e', 'd'}
  - X=set([1, 2, 3, 4])  # using constructor
  - X

    {1, 2, 3, 4}

# Set Operations

- x – y # Difference
- x | y # Union
- x & y # Intersection
- x ^ y # Symmetric difference (XOR)
- x > y, x < y # Superset, subset
- 'e' in x # Membership (sets)
- x.intersection(y) # Same as x & y
- x.union(y)

# Set Operations

- z.add(item) # Insert one item
- z.update(iterable) # Merge: in-place
- z.remove('b') # Delete one item
- z.discard('b') # Delete one item
- for item in s1:
  print(item)
- len(s1)
- s1.issubset(s2)
- s1.issuperset(s2)
- s1.symmetric_difference(s2)
- s1.pop() # removes last item, since unordered
  this is not recommended

# Set Operations

- s1.clear()  # removes all elements of set
- del s1  #deletes the set itself
- s1.copy()  # returns a copy of the set