

OBJECT ORIENTED PROGRAMMING

Lists and Dictionaries

Zuhaib A. Shaikh,

Asst. Prof., CSE Deptt., QUEST

Web: zuhaib-shaikh.neocities.org

Lists

- A data structure in python
- Similar to array in C/C++ but can carry homogenous or heterogenous data
- Index starts from zero
 - e.g. D= ['Monday', 25, 09, 2017]
- Elements can be accessed using its index
 - e.g. D[0], D[2:], etc.
- Basic operations:
 - `len(list)` – Returns length of list (no. of elements)
 - `cmp(list1, list2)` – Compare elements of 1st list with elements of 2nd list
 - `max(list)` – Returns element with maximum value
 - `min(list)` – Returns element with minimum value
 - `.index(element)` – Returns index of element from list (if it exists)
 - `list1+list2` – Returns concatenated list from list 1 and list 2, same as `.extend(sequence)`
 - `.pop()` – Returns and removes the last element of the list
 - `.append(element)` – Insert the element at the end of list
 - `.insert(index, element)` – Insert the element at the index in the list
 - `.sort(list)` – Sorts the element in the list
 - `.reverse(list)` – Reverse the elements in the list

Tuples

- Same as list but immutable
- Protects the data from accidental modification
- Small brackets are used instead of square for creating tuples
- Index starts from zero
 - e.g. D= ('Monday', 25, 09, 2017)
- Elements can be accessed using its index
 - e.g. D[0], D[2:], etc.
- Basic operations:
 - `len(tuple)` – Returns length of tuple (no. of elements)
 - `cmp(tuple1, tuple2)` – Compare elements of 1st tuple with elements of 2nd tuple
 - `max(tuple)` – Returns element with maximum value
 - `min(tuple)` – Returns element with minimum value
 - `.index(element)` – Returns index of element from tuple (if it exists)
 - `tuple1+ tuple2` – Returns concatenated list from tuple 1 and tuple 2
- List and tuples can be casted
 - `list(sequence)` and `tuple(sequence)` respectively

Dictionary

- Same as list but the index(key) and elements(items) are to be defined by programmer
- The keys and the elements are separated by : symbol
 - e.g. Info= {'name' : 'Zuhaib', 'age' : 30, 'dept' : 'CSE'}
- Elements can be accessed using its index
 - e.g. D['age'], etc.
- Basic operations:
 - `cmp(Dictionary1, Dictionary2)` – Compare elements 2 dictionary with each other
 - `len(Dictionary)` – Returns number of items in dictionary
 - `str(list)` – Returns string of keys and items
 - `.update(Dictionary2)` – Append dictionary 2 items
 - `.keys()` – Returns all keys
 - `.values()` – Returns all values
 - `.items()` – Returns list keys w.r.t their items
 - `.clear()` – Removes all keys and items
 - `.pop(key)` – Removes and return key w.r.t. item
 - `.has_key(key)` – Returns **True** if key exists



Questions

